



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI UNTUK MEMPERBAIKI KEMAMPUAN MOTORIK HALUS PADA ANAK PENDERITA AUTISME DENGAN MENGUNAKAN TEKNOLOGI LEAP MOTION CONTROLLER

Ahmad Fathoriq Fauzi
NRP 5112100123

Dosen Pembimbing
Wijayanti Nurul Khotimah, S.Kom, M.Sc.
Dr. Darlis Herumurti, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016



TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI UNTUK MEMPERBAIKI KEMAMPUAN MOTORIK HALUS PADA ANAK PENDERITA AUTISME DENGAN MENGUNAKAN TEKNOLOGI LEAP MOTION CONTROLLER

Ahmad Fathoriq Fauzi
NRP 5112100123

Dosen Pembimbing
Wijayanti Nurul Khotimah, S.Kom, M.Sc.
Dr. Darlis Herumurti, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

APPLICATION DESIGN FOR IMPROVING FINE MOTOR SKILLS IN CHILDREN WITH AUTISM USING LEAP MOTION CONTROLLER

Ahmad Fathoriq Fauzi
NRP 5112100123

Advisor
Wijayanti Nurul Khotimah, S.Kom, M.Sc.
Dr. Darlis Herumurti, S.Kom., M.Kom.

DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

**Rancang Bangun Aplikasi untuk Memperbaiki Kemampuan
Motorik Halus pada Anak Penderita Autisme dengan
Menggunakan Teknologi Leap Motion Controller**

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Interaksi Grafis dan Seni
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

Ahmad Fathoriq Fauzi

NRP : 5112100123

Disetujui oleh Dosen Pembimbing Tugas Akhir :

WIJAYANTI NURUL KHOTIMAH

S.Kom, M.Sc.

NIP: 198603122012122004

Dr. DARLIS HERUMURTI, S.Kom, M.Ts

M.Kom.

NIP: 197712172003121001



(pembimbing 1)

(pembimbing 2)

SURABAYA

JUNI 2016

[Halaman ini sengaja dikosongkan]

Rancang Bangun Aplikasi untuk Memperbaiki Kemampuan Motorik Halus pada Anak Penderita Autisme dengan Menggunakan Teknologi Leap Motion Controller

Nama Mahasiswa : Ahmad Fathoriq Fauzi
NRP : 5112100123
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing 1 : Wijayanti Nurul Khotimah, S.Kom, M.Sc.
Dosen Pembimbing 2 : Dr. Darlis Herumurti, S.Kom., M.Kom.

ABSTRAK

Autisme adalah sebuah kelainan dalam perkembangan syaraf yang ditandai dengan adanya kesulitan dalam berinteraksi sosial dan juga gangguan dalam syaraf motorik. Kebanyakan dari anak penderita autisme memiliki kelambatan dalam perkembangan motorik halus mereka, yaitu kemampuan seseorang untuk melakukan sesuatu dengan otot-otot kecil yang ada di dalam tangan. Melatih motorik halus dari anak-anak penderita autisme dilakukan dengan cara melakukan kegiatan yang memanfaatkan gerakan tangan melalui permainan ataupun kegiatan-kegiatan yang biasa dilakukan sehari-hari.

Dengan memanfaatkan Leap Motion Controller, sebuah perangkat yang menggunakan metode motion capture untuk menunjang interaksi pada antarmuka komputer yang melakukan tracking pada tangan pengguna, pada Tugas Akhir ini penulis membuat permainan yang menggunakan gerakan tangan untuk memainkannya dengan tujuan untuk melatih kemampuan motorik halus dari anak.

Uji coba pada Tugas Akhir ini menggunakan skenario uji coba dengan metode kotak hitam dan kuesioner dari pengguna.

Penguji aplikasi Tugas Akhir ini adalah anak-anak berkebutuhan khusus berumur 4-12 tahun. Dari hasil pengujian dengan kuesioner yang telah diberikan kepada guru yang mendampingi anak-anak yang melakukan uji coba aplikasi ini adalah permainan-permainan pada aplikasi ini dapat membantu anak-anak untuk fokus dan juga membantu perkembangan motorik halus dari anak.

Kata kunci: Autisme, Motorik Halus, Leap Motion Controller

Application Design for Improving Fine Motor Skills in Children with Autism using Leap Motion Controller

Student Name : Ahmad Fathoriq Fauzi
Student ID : 5112100123
Major : Teknik Informatika FTIf-ITS
Advisor 1 : Wijayanti Nurul Khotimah, S.Kom, M.Sc.
Advisor 2 : Dr. Darlis Herumurti, S.Kom., M.Kom

ABSTRACT

Autism is a complex neurodevelopmental disorder in which is characterized by difficulties in social interaction and also disturbances in motor neurons. Most of the children with autism have slower development in their fine motor development. Fine motor itself is the ability to do something with the small muscles in the hand. Delays in the development of fine motor difficulties can affect child with autism in performing their daily activities.

With Leap Motion Controller, a device using motion capture to support interaction on computer interface, which functioned to track the user's hands, can be used to train the fine motor skills of children with autism by conducting activities that take advantage of the movement of the hand, in this case by playing games that can help improve fine motor skills of children with autism.

This final project tested by using test scenarios with a black box methods and questionnaires from users. Final application testers are children with special needs aged 4-12 years. From the results of the questionnaire that was given to teachers who accompany children to test this application, games in this application can help children to focus and also help fine motor development of the child.

Keywords: Autism, Fine Motor, Leap Motion Controller

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR KODE SUMBER	xxi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Permasalahan.....	2
1.3. Batasan Permasalahan	2
1.4. Tujuan.....	3
1.5. Manfaat.....	3
1.6. Metodologi	3
1.6.1. Penyusunan Proposal Tugas Akhir.....	3
1.6.2. Studi literatur	4
1.6.3. Analisis dan Perancangan Sistem.....	4
1.6.4. Implementasi	4
1.6.5. Pengujian dan Evaluasi	4
1.6.6. Penyusunan Buku Tugas Akhir	5
1.7. Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA.....	7
2.1. Interaksi Manusia dan Komputer	7
2.2. Permodelan Tiga Dimensi (3D)	7
2.3. Leap Motion Controller	8
2.4. Leap Motion SDK	9
2.5. Visual Studio	9
2.6. Unity	10
2.7. Blender	11
2.8. Terapi untuk penderita Autisme	12
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	15

3.1.	Analisis Perangkat Lunak.....	15
3.1.1.	Deskripsi Umum Perangkat Lunak.....	15
3.1.2.	Spesifikasi Kebutuhan Perangkat Lunak.....	16
3.1.3.	Identifikasi Pengguna	16
3.2.	Perancangan Perangkat Lunak.....	17
3.2.1.	Model Kasus Penggunaan	17
3.2.2.	Definisi Kasus Penggunaan	18
3.2.3.	Definisi Aktor	23
3.2.4.	Perancangan Model 3D	24
3.2.5.	Arsitektur Umum Sistem	30
3.2.6.	Rancangan Antarmuka Aplikasi	30
3.2.7.	Perancangan Proses	35
BAB IV IMPLEMENTASI.....		45
4.1.	Lingkungan Implementasi	45
4.2.	Implementasi Antarmuka	45
4.2.1.	Implementasi Antarmuka Halaman Menu Utama.....	46
4.2.2.	Implementasi Antarmuka Halaman Permainan “Get The Ball”.....	47
4.2.3.	Implementasi Antarmuka Halaman Permainan “Match The Shape”	49
4.2.4.	Implementasi Antarmuka Halaman Permainan “Find The Matching Animals”	50
4.2.5.	Implementasi Antarmuka Halaman Permainan “Pick The Fruit”	51
4.2.6.	Implementasi Antarmuka Halaman Ketika Permainan Selesai.....	52
4.3.	Implementasi Proses	54
4.3.1.	Proses Memainkan Permainan “Get The Ball”	54
4.3.2.	Proses Memainkan Permainan “Match The Shape”	56
4.3.3.	Proses Memainkan Permainan “Find The Matching Animals”	62
4.3.4.	Proses Memainkan Permainan “Pick The Fruit”	66
BAB V PENGUJIAN DAN EVALUASI		71
5.1.	Lingkungan Pengujian.....	71

5.2.	Pengujian Aplikasi	71
5.2.1.	Skenario Pengujian Fungsionalitas.....	72
5.2.2.	Hasil Uji Coba Fungsionalitas Aplikasi	72
5.2.3.	Pengujian oleh Penulis	73
5.2.4.	Pengujian oleh anak penderita autisme	82
5.2.5.	Hasil Uji Coba	84
5.3.	Evaluasi Pengujian	90
BAB VI KESIMPULAN DAN SARAN.....		93
6.1.	Kesimpulan.....	93
6.2.	Saran.....	93
DAFTAR PUSTAKA.....		95
LAMPIRAN		97
BIODATA PENULIS.....		101

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Penggunaan Leap Motion Controller	8
Gambar 2.2 Tampilan Antarmuka Visual Studio	10
Gambar 2.3 Tampilan Antarmuka pada Unity	11
Gambar 2.4 Tampilan Antarmuka pada Blender	12
Gambar 2.5 Aktivitas pada Terapi Okupasi [10]	13
Gambar 3.1 Diagram Kasus Penggunaan	18
Gambar 3.2 Rancangan Sederhana Arsitektur Aplikasi	30
Gambar 3.3 Rancangan Antarmuka Menu Utama	31
Gambar 3.4 Rancangan Antarmuka Permainan “Get The Ball”	32
Gambar 3.5 Rancangan Antarmuka Permainan “Match The Shape”	33
Gambar 3.6 Rancangan Antarmuka Permainan “Find The Matching Animals”	33
Gambar 3.7 Rancangan Antarmuka Permainan “Pick The Fruit”	34
Gambar 3.8 Rancangan Antarmuka Permainan Selesai	35
Gambar 3.9 <i>Flowchart</i> Proses Permainan “Get The Ball”	37
Gambar 3.10 <i>Flowchart</i> Proses Permainan “Match The Shape”	39
Gambar 3.11 <i>Flowchart</i> Proses Permainan “Find The Matching Animals”	41
Gambar 3.12 <i>Flowchart</i> Proses Permainan “Pick The Fruit”	43
Gambar 4.1 Implementasi Tampilan Halaman Menu Utama	46
Gambar 4.2 Implementasi Tampilan Halaman Permainan “Get The Ball”	48
Gambar 4.3 Implementasi Tampilan Halaman Permainan “Match The Shape”	49
Gambar 4.4 Implementasi Tampilan Halaman Permainan “Find The Matching Animals”	50
Gambar 4.5 Implementasi Tampilan Halaman Permainan “Pick The Fruit”	52
Gambar 4.6 Implementasi Tampilan Halaman ketika permainan selesai	53

Gambar 5.1 Tampilan Saat Uji Coba Memainkan Permainan "Get The Ball"	75
Gambar 5.2 Tampilan Saat Uji Coba Memainkan Permainan "Match The Shape"	77
Gambar 5.3 Tampilan Saat Uji Coba Memainkan Permainan "Find The Matching Animals"	80
Gambar 5.4 Tampilan Saat Uji Coba Memainkan Permainan "Pick The Fruit"	82
Gambar 5.5 Grafik performa bermain anak ketiga	88
Gambar A.1 Kuesioner Responden Pertama	97
Gambar A.2 Kuesioner Responden Kedua	98
Gambar A.3 Kuesioner Responden Ketiga	99
Gambar A.4 Kuesioner Responden Keempat	100

DAFTAR TABEL

Tabel 3.1 Karakteristik Pengguna	17
Tabel 3.2 Daftar Kasus penggunaan.....	17
Tabel 3.3 Spesifikasi Kasus Memainkan Permainan “Get The Ball”	18
Tabel 3.4 Spesifikasi Kasus Memainkan Permainan “Match The Shape”	19
Tabel 3.5 Spesifikasi Kasus Penggunaan Memainkan Permainan “Find The Matching Animals”	21
Tabel 3.6 Spesifikasi Kasus Memainkan Permainan “Pick The Fruit”	22
Tabel 3.7 Deskripsi Pengguna.....	24
Tabel 3.8 Daftar Objek 3D dari <i>Standard Asset</i> Unity.....	24
Tabel 3.9 Daftar Objek 3D hasil unduhan.....	25
Tabel 3.10 Daftar Objek 3D	29
Tabel 5.1 Lingkungan Pengujian Sistem.....	71
Tabel 5.2 Hasil Uji Coba Memainkan Permainan “Get The Ball”	73
Tabel 5.3 Hasil Uji Coba Memainkan Permainan “Match The Shape”	75
Tabel 5.4 Hasil Uji Coba Memainkan Permainan “Find The Matching Animals”	78
Tabel 5.5 Hasil Uji Coba Memainkan Permainan “Pick The Fruit”	80
Tabel 5.6 Kuesioner Pengguna.....	83
Tabel 5.7 Hasil Kuesioner	84
Tabel 5.8 Data Performa Pemain Permainan “Get The Ball”	86
Tabel 5.9 Data Performa Pemain Permainan “Match The Shape”	86
Tabel 5.10 Data Performa Pemain Permainan “Find The Matching Animals”	86
Tabel 5.11 Data Performa Pemain Permainan “Pick The Fruit”	87

Tabel 5.12 Data Performa Anak ke-3 pada Skenario 3	88
Tabel 5.13 Tanggapan Pengguna.....	90

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Tampilan Halaman Menu Utama	47
Kode Sumber 4.2 Implementasi Antarmuka Halaman Permainan “Get The Ball”	48
Kode Sumber 4.3 Implementasi Antarmuka Halaman Permainan “Match The Shape”	50
Kode Sumber 4.4 Implementasi Antarmuka Halaman Permainan “Find The Matching Animals”	51
Kode Sumber 4.5 Implementasi Antarmuka Halaman Permainan “Pick The Fruit”	52
Kode Sumber 4.6 Implementasi Antarmuka Halaman ketika permainan selesai.....	54
Kode Sumber 4.7 Memainkan Permainan “Get The Ball”	56
Kode Sumber 4.8 Memainkan Permainan Match The Shape.....	62
Kode Sumber 4.9 Memainkan Permainan “Find The Matching Animals”	65
Kode Sumber 4.10 Memainkan Permainan “Pick The Fruit”	70

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

1.1. Latar Belakang

Autisme adalah sebuah kelainan dalam perkembangan syaraf yang ditandai dengan adanya kesulitan dalam berinteraksi sosial dan juga gangguan dalam syaraf motorik. Kondisi ini dapat mengenai seseorang sejak lahir ataupun pada masa balita, yang membuat dirinya tidak dapat membangun hubungan sosial atau komunikasi dengan normal [1].

Kebanyakan dari anak penderita autisme memiliki kelambatan dalam perkembangan motorik halus mereka. Motorik halus ini sendiri adalah kemampuan seseorang untuk melakukan sesuatu dengan otot-otot kecil yang ada di dalam tangan, gerakan mereka kaku dan kasar sehingga menyulitkan mereka dalam memegang pensil dengan benar, sulit dalam mengancingkan baju dan lain sebagainya. Meningkatkan kemampuan motorik dari anak penderita autisme merupakan hal yang penting untuk dilakukan. Hal ini bertujuan untuk meningkatkan kemampuan belajar dan memperbaiki koordinasi pada anak penderita autisme [2].

Terapi untuk melatih keterampilan motorik halus sudah sering dipraktikkan, seperti pada percobaan yang dilakukan oleh Y. Chang [3] yang menerapkan terapi keterampilan motorik halus dengan menggunakan Kinect, namun Kinect lebih terfokus pada mengenali pergerakan seseorang dan juga pada titik-titik dari tulang. Padahal dalam melatih kemampuan motorik halus dari seseorang, lebih baik difokuskan dalam melatih otot-otot halus yang ada pada tangan, tidak perlu terlalu fokus ke gerakan tubuh atau pun titik-titik tulang. Dengan menggunakan Leap Motion, sebuah perangkat yang menggunakan metode *motion capture*

untuk menunjang interaksi pada antarmuka komputer yang hanya melakukan *tracking* pada tangan pengguna, diharapkan akan didapatkan akurasi yang lebih baik daripada dengan menggunakan Kinect yang melakukan *tracking* pada seluruh tubuh.

Maka dari itu, penulis akan membuat aplikasi permainan yang bertujuan untuk melatih motorik halus dari pengguna yang lebih dikhususkan lagi pada anak-anak yang menderita autisme. Penulis menggunakan teknologi Leap Motion untuk menjadi *controller* dari permainan yang akan dapat dimainkan pada komputer. Dengan adanya aplikasi ini diharapkan dapat menjadi salah satu alat pembelajaran dan juga terapi alternatif pada anak-anak penderita autisme.

1.2. Rumusan Permasalahan

Rumusan masalah yang diangkat pada Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana merancang jenis-jenis permainan yang dapat melatih kemampuan motorik halus dari pengguna?
2. Bagaimana merancang sebuah aplikasi permainan yang dapat melatih kemampuan motorik halus dengan memanfaatkan teknologi Leap Motion Controller?

1.3. Batasan Permasalahan

Beberapa batasan masalah yang terdapat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Aplikasi yang dibuat merupakan aplikasi *desktop*.
2. Lingkungan pengembangan yang digunakan menggunakan aplikasi Unity dan bahasa pemrograman C#.
3. Menggunakan Leap Motion SDK.
4. Permainan yang dilakukan:
 - Get The Ball: Memindahkan bola ke ember yang berwarna sama.
 - Match the Shape: Memilih bentuk bangun datar yang sama dari gambar yang diberikan.

- Find the Matching Animals: Memasangkan binatang yang sama dari beberapa binatang yang muncul.
 - Pick the Fruit: Mengambil buah dari pohon nya ke keranjang yang sesuai dengan gambar yang diberikan.
5. Permainan dimainkan menggunakan tangan melalui teknologi Leap Motion Controller.

1.4. Tujuan

Tujuan dari pembuatan Tugas Akhir ini antara lain:

1. Pemanfaatan dan pengembangan Leap Motion untuk membuat permainan yang bermanfaat.
2. Membuat permainan yang dapat melatih kemampuan motorik halus dari pengguna.

1.5. Manfaat

Tugas Akhir ini diharapkan mampu memberikan sebuah permainan alternatif dalam terapi untuk anak-anak penderita autisme dalam rangka melatih kemampuan motorik halus mereka.

1.6. Metodologi

Pembuatan Tugas Akhir dilakukan menggunakan metodologi sebagai berikut.

1.6.1. Penyusunan Proposal Tugas Akhir

Proposal Tugas Akhir ini berisi tentang deskripsi pendahuluan dari Tugas Akhir yang akan dibuat. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya usulan Tugas Akhir, rumusan masalah yang diangkat, batasan masalah untuk Tugas Akhir, tujuan dari pembuatan Tugas Akhir, dan manfaat dari hasil pembuatan Tugas Akhir. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan Tugas Akhir. Subbab metodologi berisi penjelasan mengenai tahapan penyusunan Tugas Akhir mulai dari penyusunan

proposal hingga penyusunan buku Tugas Akhir. Terdapat pula subbab jadwal kegiatan yang menjelaskan jadwal pengerjaan Tugas Akhir.

1.6.2. Studi literatur

Pada studi literatur ini, akan dipelajari tentang hal-hal yang dibutuhkan sebagai referensi dalam pembuatan aplikasi yaitu mengenai Leap Motion Controller, Leap Motion SDK, Unity, dan juga Terapi Okupasi. Informasi dan studi literatur tersebut didapat dari buku, internet, dan materi-materi kuliah yang berhubungan dengan metode yang digunakan.

1.6.3. Analisis dan Perancangan Sistem

Adapun fitur yang terdapat dalam aplikasi ini nantinya antara lain:

1. “Get the Ball”.
2. “Match the Shape”.
3. “Find the Matching Animals”.
4. “Pick the Fruit”.

1.6.4. Implementasi

Pada tahap ini dilakukan implementasi permodelan objek-objek 3D dan implementasi proses yang telah didefinisikan pada bab Analisis dan Perancangan Sistem. Kemudian dilakukan integrasi aplikasi dengan perangkat Leap Motion. Aplikasi ini dibangun dengan Unity dan Blender.

1.6.5. Pengujian dan Evaluasi

Pada tahap ini dilakukan pengujian terhadap aplikasi. Pengujian akan dilakukan kepada pengguna, yaitu beberapa anak penderita autisme berumur 4 - 12 tahun. Hasil pengujian aplikasi akan diberikan evaluasi serta ditarik kesimpulan untuk kemudian

diujikan kembali untuk memastikan aplikasi telah berfungsi sebagaimana mestinya.

1.6.6. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam Tugas Akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat.

1.7. Sistematika Penulisan

Buku Tugas Akhir ini terdiri dari beberapa bab, yang dijelaskan sebagai berikut:

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, rumusan dan batasan permasalahan, tujuan dan manfaat pembuatan Tugas Akhir, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Tinjauan Pustaka

Bab ini membahas dasar pembuatan dan beberapa teori penunjang yang berhubungan dengan pokok pembahasan yang mendasari pembuatan Tugas Akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas analisis dari sistem yang dibuat meliputi analisis permasalahan, deskripsi umum perangkat lunak, spesifikasi kebutuhan, dan identifikasi pengguna. Kemudian membahas rancangan dari sistem yang dibuat meliputi rancangan skenario kasus penggunaan, arsitektur, data, dan antarmuka.

Bab IV Implementasi

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat keluaran yang dihasilkan oleh aplikasi dan evaluasi untuk mengetahui kemampuan aplikasi.

Bab VI Penutup

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan serta saran untuk pengembangan aplikasi selanjutnya.

BAB II

TINJAUAN PUSTAKA

Bab ini menjelaskan tentang tinjauan pustaka yang menjadi dasar pembuatan Tugas Akhir. Beberapa teori, pustaka, dan teknologi yang mendasari pengerjaan Tugas Akhir ini di antaranya meliputi definisi permodelan tiga dimensi, Leap Motion Controller, SDK Leap Motion, Blender, Visual Studio, dan juga terapi-terapi yang dilakukan oleh penderita autisme. Penjelasan secara khusus masing-masing tinjauan pustaka dapat dilihat pada masing-masing subbab berikut ini.

2.1. Interaksi Manusia dan Komputer

Interaksi manusia dan komputer adalah disiplin ilmu yang mempelajari hubungan antara manusia dan komputer yang meliputi perancangan, pengevaluasian, dan pengimplementasian antarmuka pengguna komputer agar mudah digunakan oleh manusia. Ilmu ini berusaha menemukan cara yang paling efisien untuk merancang pesan elektronik [4].

Sedangkan interaksi manusia dan komputer sendiri adalah serangkaian proses, dialog dan kegiatan yang dilakukan oleh manusia untuk berinteraksi dengan komputer yang keduanya saling memberikan masukan dan umpan balik melalui sebuah antarmuka untuk memperoleh hasil akhir yang diharapkan.

2.2. Permodelan Tiga Dimensi (3D)

Permodelan adalah membentuk suatu benda-benda atau objek. Membuat dan mendesain objek tersebut sehingga terlihat seperti hidup. Sesuai dengan objek dan basisnya, proses ini secara keseluruhan dikerjakan di komputer. Melalui konsep dan proses desain, keseluruhan objek bisa diperlihatkan secara tiga dimensi, sehingga banyak yang menyebut hasil ini sebagai permodelan tiga dimensi [5].

Permodelan tiga dimensi memiliki beberapa aspek yang harus diperhatikan, yaitu pendeskripsian objek, tujuan dari model, tingkat kerumitan, kesesuaian dan kenyamanan, serta kemudahan manipulasi objek. Permodelan tiga dimensi dapat dilakukan dengan bantuan aplikasi-aplikasi, seperti Blender, K-3D, Google SketchUp, dan masih banyak lainnya.

2.3. Leap Motion Controller

Leap Motion Controller adalah sebuah perangkat yang dikembangkan oleh perusahaan Leap Motion, Inc yang dikembangkan sejak tahun 2008 oleh David Holz. Penggunaan Leap Motion Controller biasanya digunakan sebagai *input* dari komputer tanpa sentuh, bisa dikatakan bahwa Leap Motion ini merupakan pengganti *mouse*, karena mempunyai tujuan dan fungsi yang sama [6].



Gambar 2.1 Penggunaan Leap Motion Controller

Leap Motion menggunakan sinar inframerah yang dipancarkan untuk mendeteksi bayangan dari sinar yang dihasilkan dengan kamera inframerah. Jangkauan yang dapat dideteksi oleh Leap Motion Controller adalah sejauh 150°. Penggunaan Leap Motion Controller dapat dilihat pada Gambar 2.1.

2.4. Leap Motion SDK

Leap Motion SDK adalah sekumpulan *library* yang berisikan tentang kebutuhan - kebutuhan sistem dari aplikasi yang digunakan untuk mengembangkan Leap Motion Controller ini, biasanya terdapat beberapa fungsi yang dapat digunakan seperti permodelan tangan dan jari - jari manusia. Leap Motion SDK ini di buat untuk memudahkan developer dalam membangun aplikasi. Pengembang dapat mengunduhnya secara gratis dan tersedia dalam berbadai jenis bahasa pemrograman yang berbeda. Antara lain adalah Javascript, Unity / C#, C++, Java, Python, dan Objective-C. Juga mendukung berbagai macam sistem operasi yang berbeda yakni Windows, OSX, dan Linux [6].

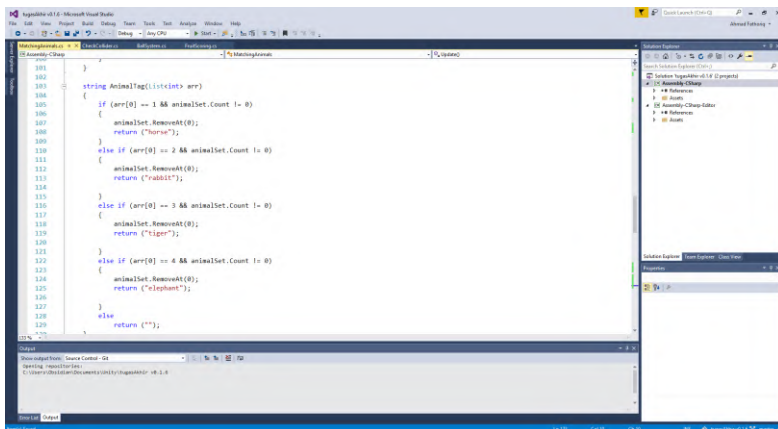
Pada tiap versi-versi dari Leap Motion SDK pun terdapat perbedaan signifikan yang terhadap fitur-fitur yang disediakan. Contohnya saja pada versi 2 Leap Motion SDK terdapat banyak gestur-gestur tambahan seperti *magnetic pitch*, *swipe*, *slide* dan lain-lain, namun pada versi 3 Leap Motion SDK tidak terdapat terlalu banyak tambahan gestur namun SDK ini lebih dipermudah untuk pengguna yang ingin menggunakan SDK untuk membuat aplikasi *virtual reality*. Versi terbaru Leap Motion ini dinamakan dengan Leap Motion ORION.

2.5. Visual Studio

Microsoft Visual Studio merupakan sebuah aplikasi yang terdiri dari *compiler*, SDK, *integrated development environment (IDE)*, dan dokumentasi yang biasa digunakan untuk mengembangkan aplikasi pada *personal computer*, aplikasi *web*, aplikasi *mobile*, aplikasi *console* maupun komponen aplikasi pada Microsoft Visual Studio. Banyak *compiler* yang disediakan oleh

Microsoft Visual Studio, misalnya saja Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe [7].

Para pengembang aplikasi banyak yang mengembangkan aplikasinya menggunakan Microsoft Visual Studio karena banyak fungsi-fungsi tambahan yang telah disediakan oleh Microsoft Visual Studio yang diperuntukkan untuk pengembang agar lebih cepat dalam pengembangan aplikasi mereka dan sehingga pengembang lebih mudah dan juga terbantu dalam mengembangkan maupun membuat aplikasi. Tampilan antarmuka aplikasi ini dapat dilihat pada Gambar 2.2.



Gambar 2.2 Tampilan Antarmuka Visual Studio

2.6. Unity

Unity adalah sebuah produk berupa game engine yang dikembangkan oleh Unity Technologies, Unity sendiri dapat menciptakan game ke dalam beberapa sistem operasi sekaligus. Antara lain: Windows Phone, Android, iOS, Windows 8, OSX, Blackberry 10, Playstation 3, Playstation 4, XBOX, dan sebagainya. Game yang dapat dibuat oleh Unity ini bisa dalam bentuk 3D ataupun 2D, tergantung pada pengembang game tersebut. Unity mampu untuk mengubah gambar statis menjadi

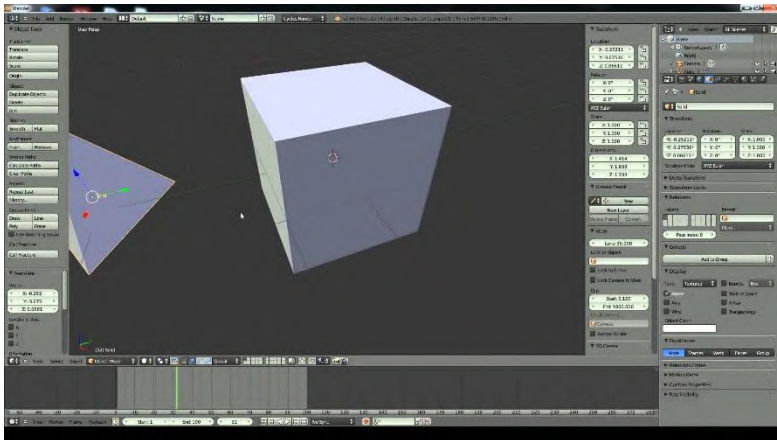
animasi yang dapat dimainkan [8]. Tampilan antarmuka aplikasi ini dapat dilihat pada Gambar 2.3.



Gambar 2.3 Tampilan Antarmuka pada Unity

2.7. Blender

Blender adalah sebuah perangkat lunak yang berfungsi untuk melakukan permodelan tiga dimensi. Blender juga dapat digunakan untuk membuat animasi tiga dimensi. Blender tersedia dapat diunduh pada situs <https://www.blender.org/> dengan gratis atau tanpa dikenakan biaya untuk mengunduhnya [9]. Blender dapat digunakan pada berbagai jenis sistem operasi. Keluaran dari Blender adalah objek-objek tiga dimensi dengan berbagai format, seperti .obj, .3ds, dan lain-lain. Objek-objek yang dibuat dengan menggunakan dapat digunakan sebagai material dasar pembuatan aplikasi atau permainan dengan menggunakan game engine atau perangkat lunak lain yang dapat digunakan untuk membuat aplikasi tiga dimensi. Tampilan antarmuka aplikasi ini dapat dilihat pada Gambar 2.4.



Gambar 2.4 Tampilan Antarmuka pada Blender

2.8. Terapi untuk penderita Autisme

Autisme merupakan sebuah gangguan yang mengakibatkan anak-anak yang menderitanya ini tertinggal dengan anak-anak yang lain dalam hal pemahaman dalam pembelajaran yang diberikan oleh guru di sekolah, hal ini sebenarnya dapat diperbaiki jika diberikan penanganan yang baik.

Terdapat banyak jenis terapi yang telah dikembangkan untuk mengembangkan kemampuan anak penderita autisme seperti terapi perilaku yang diperuntukkan untuk meningkatkan pemahaman dan kepatuhan anak pada aturan, terapi wicara yang diperuntukkan untuk meningkatkan kemampuan bicaranya untuk berkomunikasi/berinteraksi dengan orang lain, terapi fisik yang diperuntukkan untuk menguatkan otot-otot dan memperbaiki keseimbangan tubuh anak, terapi okupasi yang diperuntukkan untuk pelatihan untuk membuat semua otot dalam tubuhnya berfungsi dengan tepat dan juga terapi-terapi lainnya. Dengan mengikuti terapi-terapi tersebut penderita autisme dapat terbantu untuk lebih mandiri dan juga dapat menyerap informasi atau materi pada sekolah layaknya anak-anak pada normalnya [2].

Dalam hal perkembangan motorik halus dari penderita autisme dilakukan terapi okupasi. Terapi okupasi ini bergerak pada tiga area yaitu yang pertama kegiatan sehari-hari (*activity of daily living*) seperti aktivitas mandi, makan, dan berpakaian, yang kedua adalah produktivitas, dan yang terakhir adalah pemanfaatan waktu luang (*leisure activity*) yang di mana ketiga area ini yang merupakan tiga komponen yang diperlukan bagi tiap individu.



Gambar 2.5 Aktivitas pada Terapi Okupasi [10]

Terapi okupasi biasanya dapat dilakukan secara perorangan ataupun kelompok, bentuk kegiatannya dibagi menjadi dua yaitu *activity day learning* (ADL) dan permainan. ADL dilakukan dengan memberikan keterampilan hidup lebih mandiri dan terampil. Misalnya seperti latihan menyimpul sepatu, dan mengancingkan baju, aktivitas pada terapi okupasi dapat dilihat pada Gambar 2.5. Sedangkan permainan ini merupakan terapi yang berbentuk bermain untuk memberikan kesenangan dan sosialisasi yang baik. Misalnya dengan bermain lempar bola dan bermain tebak kata.

Tujuan dari terapi okupasi ini adalah salah satunya sebagai hiburan/kesenangan agar dapat mengurangi rasa rendah diri dan memupuk semangat kerja anak. Selain itu terapi ini juga dilakukan untuk membantu menguatkan, memperbaiki koordinasi, dan keterampilan otot pada anak penderita autisme dengan kata lain untuk melatih motorik halus dari anak penderita autisme [2].

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan tentang analisis dan perancangan aplikasi permainan untuk anak penderita autisme menggunakan teknologi Leap Motion Controller (LMC). Pembahasan yang akan dilakukan meliputi analisis fitur yang dibutuhkan dan perancangan perangkat lunak.

3.1. Analisis Perangkat Lunak

Subbab ini menjelaskan tentang hasil analisis kebutuhan perangkat lunak serta arsitektur aplikasi permainan untuk anak penderita autisme menggunakan teknologi LMC. Tiap-tiap subbab menjelaskan tentang deskripsi umum perangkat lunak, spesifikasi kebutuhan perangkat lunak, analisis aktor, arsitektur perangkat lunak, dan skenario kasus penggunaan.

3.1.1. Deskripsi Umum Perangkat Lunak

Pada Tugas Akhir ini dibuat sebuah aplikasi permainan untuk anak penderita autisme menggunakan teknologi LMC merupakan sebuah aplikasi yang bertujuan untuk meningkatkan keterampilan motorik halus dengan melakukan permainan-permainan sederhana yang diinspirasi oleh terapi okupasi. Hal ini dilakukan dengan menggunakan LMC dalam memainkan permainan-permainan yang ada. Gerakan-gerakan serta tema dari permainan disesuaikan agar dapat meningkatkan motorik halus dari anak-anak penderita autisme.

Terdapat empat jenis permainan pada aplikasi ini. Permainan pertama adalah “Get The Ball” yang merupakan sebuah permainan memasukkan bola kepada ember yang sesuai dengan warna dari bola. Permainan kedua adalah “Match the Shape”, permainan memasang bentuk bangun dasar yang sama dengan bentuk yang telah ditentukan. Setelah itu ada permainan “Find The Matching Animals” di mana pengguna diminta untuk mencari dua pasang binatang yang sama dari empat binatang yang ada. Permainan

terakhir adalah “Pick the Fruit” yaitu sebuah permainan yang bertujuan mengambil buah yang sesuai dengan yang telah ditentukan.

Selain dari meningkatkan motorik halus dari anak-anak penderita autisme, diharapkan aplikasi ini pun dapat membantu mengedukasi anak-anak tersebut pada warna, bentuk-bentuk bangun dasar, bentuk-bentuk binatang serta bentuk-bentuk dari buah. Dalam tiap permainan pengguna akan dinilai berdasarkan skor dan juga lama penyelesaian dari tiap permainan. Pengguna akan memainkan tiap permainan hingga mencapai skor 5. Jika skor sudah mencapai 5 permainan akan berhenti dan waktu penyelesaian akan dicatat. Skor dan waktu penyelesaian ini akan menjadi tolak ukur kemampuan motorik halus dari pengguna. Skor dan waktu penyelesaian ini pun dapat dijadikan sebagai tolak ukur perkembangan dari kemampuan motorik halus pengguna dari waktu ke waktu.

3.1.2. Spesifikasi Kebutuhan Perangkat Lunak

Pada sistem ini, terdapat beberapa kebutuhan fungsional yang mendukung untuk jalannya aplikasi. Fungsi yang terdapat dalam aplikasi ini adalah sebagai berikut.

- a. Memainkan empat permainan dengan menggunakan LMC
Aplikasi dapat menampilkan keempat permainan yang ada dan pengguna dapat menggunakan tangan mereka dalam permainan dengan menggunakan perangkat LMC.
- b. Merekam Waktu dan Skor dari pengguna
Aplikasi memiliki fungsionalitas merekam waktu dan skor setiap pengguna yang memainkan tiap permainan yang ada dalam permainan ini.

3.1.3. Identifikasi Pengguna

Berdasarkan deskripsi umum di atas, maka dapat diketahui bahwa pengguna yang akan menggunakan aplikasi ini hanya satu orang, yaitu pengguna yang menjalankan aplikasi. Karakteristik pengguna tercantum dalam Tabel 3.1.

Tabel 3.1 Karakteristik Pengguna

Nama Aktor	Tugas	Hak Akses Aplikasi	Kemampuan yang harus dimiliki
Pengguna	Memainkan permainan yang ada pada aplikasi	Menjalankan aplikasi	Tidak ada

3.2. Perancangan Perangkat Lunak

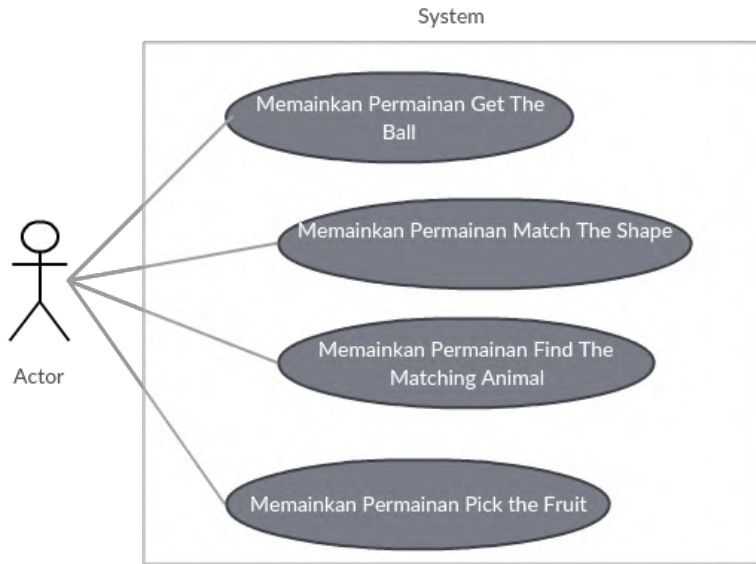
Subbab ini membahas bagaimana rancangan dari aplikasi Tugas Akhir ini. Meliputi: Model Kasus Penggunaan, Definisi Aktor, Definisi Kasus Penggunaan, Arsitektur Umum Sistem, Rancangan Antarmuka Aplikasi, dan Rancangan Proses Aplikasi.

3.2.1. Model Kasus Penggunaan

Berdasarkan analisis spesifikasi kebutuhan fungsional dan analisis aktor dari sistem dibuat kasus penggunaan sistem. Kasus-kasus penggunaan dalam sistem ini akan dijelaskan secara rinci pada subbab ini. Kasus penggunaan digambarkan dalam sebuah diagram kasus penggunaan. Diagram kasus penggunaan dapat dilihat pada Tabel 3.2 dan Gambar 3.1.

Tabel 3.2 Daftar Kasus penggunaan

Kode Kasus Penggunaan	Nama
UC-001	Memainkan permainan “Get The Ball”
UC-002	Memainkan permainan “Match The Shape”
UC-003	Memainkan permainan “Find The Matching Animals”
UC-004	Memainkan permainan “Pick The Fruit”



Gambar 3.1 Diagram Kasus Penggunaan

3.2.2. Definisi Kasus Penggunaan

Detail mengenai kasus penggunaan tersebut dapat dilihat pada subbab berikut ini.

3.2.2.1. Memainkan Permainan “Get The Ball”

Spesifikasi kasus penggunaan memainkan permainan “Get The Ball” menggunakan LMC dapat dilihat pada Tabel 3.3.

Tabel 3.3 Spesifikasi Kasus Memainkan Permainan “Get The Ball”

Nama	Memainkan permainan “Get The Ball”
Kode	UC-001
Deskripsi	Pengguna dapat memainkan permainan yang memiliki tujuan memasukkan bola pada ember yang memiliki warna sama dengan ember yang ada.

Aktor	Pengguna
Kondisi Awal	Aplikasi berada pada menu awal pada <i>homescreen</i> . Pengguna sudah menyambungkan Leap Motion dengan aplikasi.
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna menekan tombol “Get The Ball”. 2. Aplikasi akan menampilkan objek bola dan tiga ember yang berwarna berbeda pada monitor. 3. Pengguna akan ditampilkan panduan cara bermain dari permainan pada monitor. 4. Setelah panduan bermain hilang dari monitor pemain dapat mulai bermain dengan memasukkan tangan mereka ke area LMC.
-Kejadian Alternatif	A1. Pengguna menekan tombol EXIT 1. Aplikasi akan keluar dari permainan. A2. Pengguna menekan tombol HOME 2. Aplikasi akan menampilkan halaman menu utama.
Kondisi Akhir	Setelah mencapai skor tertentu atau sudah mencapai waktu yang ditentukan permainan akan berakhir.

3.2.2.2. Memainkan Permainan “Match The Shape”

Spesifikasi kasus penggunaan memainkan permainan “Match The Shape” menggunakan LMC dapat dilihat pada Tabel 3.4.

Tabel 3.4 Spesifikasi Kasus Memainkan Permainan “Match The Shape”

Nama	Memainkan permainan “Match The Shape”
-------------	---------------------------------------

Kode	UC-002
Deskripsi	Pengguna dapat memainkan permainan yang memiliki tujuan meletakkan bentuk bangun dasar yang berbentuk sama dari bangun dasar yang ditunjukkan oleh sistem.
Aktor	Pengguna
Kondisi Awal	Aplikasi berada pada menu awal pada <i>homescreen</i> . Pengguna sudah menyambungkan Leap Motion dengan aplikasi.
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna menekan tombol “Match The Shape”. 2. Aplikasi akan menampilkan objek delapan bangun ruang berbeda dan satu bangun ruang yang sama dengan salah satu bangun dasar dari delapan bangun dasar sebelumnya ada pada monitor. 3. Pengguna akan ditampilkan panduan cara bermain dari permainan pada monitor. 4. Setelah panduan bermain hilang dari monitor Pemain dapat mulai bermain dengan memasukkan tangan mereka ke area LMC.
- Kejadian Alternatif	A1. Pengguna menekan tombol EXIT <ol style="list-style-type: none"> 1. Aplikasi akan keluar dari permainan. A2. Pengguna menekan tombol HOME <ol style="list-style-type: none"> 2. Aplikasi akan menampilkan halaman menu utama.
Kondisi Akhir	Setelah mencapai skor tertentu atau sudah mencapai waktu yang ditentukan permainan akan berakhir.

3.2.2.3. Memainkan Permainan “Find The Matching Animals”

Spesifikasi kasus penggunaan memainkan permainan “Find The Matching Animals” menggunakan LMC dapat dilihat pada Tabel 3.5.

Tabel 3.5 Spesifikasi Kasus Penggunaan Memainkan Permainan “Find The Matching Animals”

Nama	Memainkan permainan “Find The Matching Animals”
Kode	UC-003
Deskripsi	Pengguna dapat memainkan permainan yang memiliki tujuan memindahkan dua binatang yang sama dari empat model binatang yang ditampilkan ke dua kandang yang telah disediakan.
Aktor	Pengguna
Kondisi Awal	Aplikasi berada pada menu awal pada <i>homescreen</i> . Pengguna sudah menyambungkan Leap Motion dengan aplikasi.
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna menekan tombol “Find The Matching Animals”. 2. Aplikasi akan menampilkan objek empat binatang dengan dua tiga binatang berbeda dan satu binatang yang sama dari tiga binatang yang ada, selain binatang kandang untuk meletakkan binatang pun akan ditampilkan pada monitor. 3. Pengguna akan ditampilkan panduan cara bermain dari permainan pada monitor. 4. Setelah panduan bermain hilang dari monitor pemain dapat mulai bermain

	dengan memasukkan tangan mereka ke area LMC.
-Kejadian Alternatif	A1. Pengguna menekan tombol EXIT 1. Aplikasi akan keluar dari permainan. A2. Pengguna menekan tombol HOME 2. Aplikasi akan menampilkan halaman menu utama.
Kondisi Akhir	Setelah mencapai skor tertentu atau sudah mencapai waktu yang ditentukan permainan akan berakhir.

3.2.2.4. Memainkan Permainan “Pick The Fruit”

Spesifikasi kasus penggunaan memainkan permainan “Pick The Fruit” menggunakan LMC dapat dilihat pada Tabel 3.6.

Tabel 3.6 Spesifikasi Kasus Memainkan Permainan “Pick The Fruit”

Nama	Memainkan permainan “Pick The Fruit”
Kode	UC-004
Deskripsi	Pengguna dapat memainkan permainan yang memiliki tujuan meletakkan bentuk buah yang berbentuk sama dari bentuk buah yang telah ditunjukkan oleh sistem.
Aktor	Pengguna
Kondisi Awal	Aplikasi berada pada menu awal pada <i>homescreen</i> . Pengguna sudah menyambungkan Leap Motion dengan aplikasi.

Nama	Memainkan permainan “Pick The Fruit”
Kode	UC-004
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna menekan tombol “Pick The Fruit”. 2. Aplikasi akan menampilkan objek empat binatang dengan dua tiga binatang berbeda dan satu binatang yang sama dari tiga binatang yang ada, selain binatang kandang untuk meletakkan binatang pun akan ditampilkan pada monitor. 3. Pengguna akan ditampilkan panduan cara bermain dari permainan pada monitor. 4. Setelah panduan bermain hilang dari monitor pemain dapat mulai bermain dengan memasukkan tangan mereka ke area LMC.
- Kejadian Alternatif	<p>A1. Pengguna menekan tombol EXIT</p> <ol style="list-style-type: none"> 1. Aplikasi akan keluar dari permainan. <p>A2. Pengguna menekan tombol HOME</p> <ol style="list-style-type: none"> 2. Aplikasi akan menampilkan halaman menu utama.
Kondisi Akhir	Setelah mencapai skor tertentu atau sudah mencapai waktu yang ditentukan permainan akan berakhir.

3.2.3. Definisi Aktor

Aktor yang terdapat dalam aplikasi permainan untuk anak penderita autisme menggunakan teknologi LMC terlihat pada Tabel 3.7.

Tabel 3.7 Deskripsi Pengguna

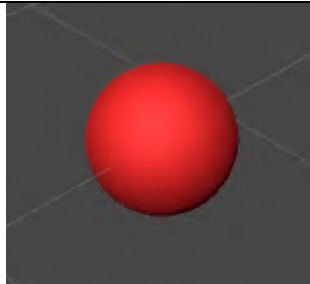
No.	Nama	Deskripsi
1	Pengguna	<ul style="list-style-type: none"> - Merupakan aktor yang bertugas untuk memainkan semua permainan pada aplikasi ini. Seluruh fungsionalitas yang ada di dalam sistem berhak digunakan oleh pengguna. - Aktor harus paham terlebih dahulu mengenai jalannya aplikasi.

3.2.4. Perancangan Model 3D

Pada aplikasi ini model 3D yang digunakan terbagi tiga, model diambil dari *standard asset* Unity, model yang diunduh dari *website* lain yang menyediakan model 3D dengan gratis dan juga yaitu model yang dari *asset store* dan selanjutnya dimodifikasi menggunakan Blender. Model-model 3D yang digunakan dalam aplikasi tersebut akan ditunjukkan pada Tabel 3.8 - Tabel 3.10.

3.2.4.1. Objek 3D dari *Standard Asset* Unity




Tabel 3.8 Daftar Objek 3D dari *Standard Asset* Unity

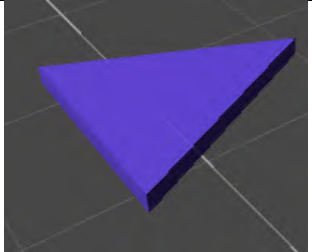
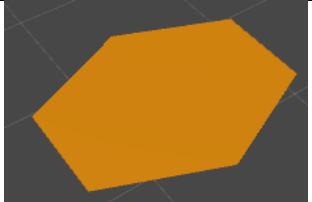
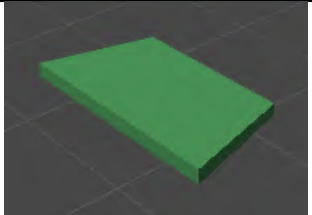
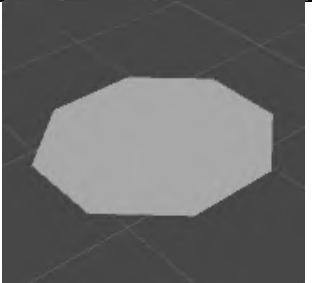
No.	Nama Objek 3D	Gambar
1.	Bola	

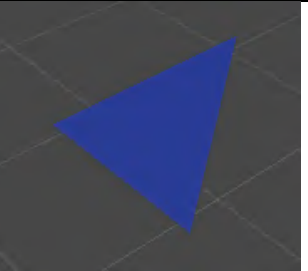



No.	Nama Objek 3D	Gambar
2.	Jeruk	

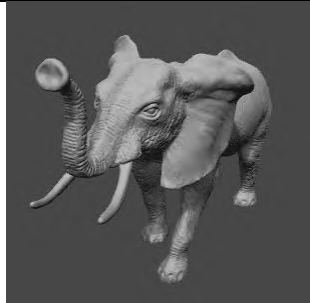
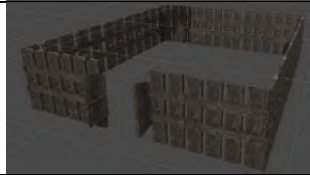


3.2.4.2. Objek 3D hasil unduhan

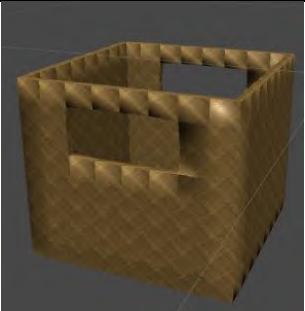
Tabel 3.9 Daftar Objek 3D hasil unduhan

No.	Nama Objek 3D	Gambar
1.	Ember (dari <i>Unity Assets Store</i>)	
2.	Bangun Ruang Bintang (dari yobi3d.com)	
3.	Bangun Ruang Segi Lima (dari yobi3d.com)	

No.	Nama Objek 3D	Gambar
4.	Bangun Ruang Segitiga Sama Kaki (dari yobi3d.com)	
5.	Bangun Ruang Segi Enam (dari yobi3d.com)	
6.	Bangun Ruang Trapesium (dari yobi3d.com)	
7.	Bangun Ruang Segi Delapan (dari yobi3d.com)	

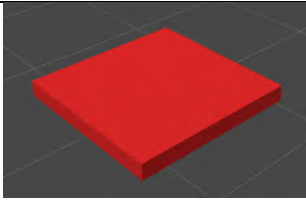
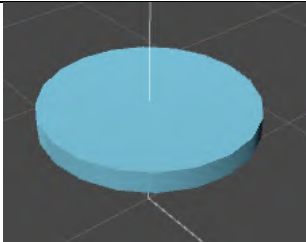
No.	Nama Objek 3D	Gambar
8.	Bangun Ruang Segitiga Siku-siku (dari yobi3d.com)	
9.	Kuda (dari tf3dm.com)	
10.	Kelinci (dari tf3dm.com)	
11.	Harimau (dari tf3dm.com)	

No.	Nama Objek 3D	Gambar
12.	Gajah (dari yobi3dcom)	 A 3D rendered model of an elephant, shown in profile facing left. It has large ears, a thick trunk curled slightly, and small tusks. The model is set against a dark gray background.
13.	Kandang (dari tf3dm.com)	 A 3D rendered model of a pigsty, constructed from dark, textured blocks. It features a rectangular enclosure with a small entrance on the right side. The model is set against a dark gray background.
17.	Apel (dari tf3dm.com)	 A 3D rendered model of a single apple, showing a gradient of red and yellow. It has a short brown stem and a small leaf. The model is set against a dark gray background with a faint grid pattern.
18.	Pir (dari tf3dm.com)	 A 3D rendered model of a single pear, showing a gradient of green and yellow. It has a short brown stem. The model is set against a dark gray background with a faint grid pattern.

No.	Nama Objek 3D	Gambar
19.	Keranjang (dari tf3dm.com)	

3.2.4.3. Objek 3D Hasil Modifikasi menggunakan aplikasi Blender

Tabel 3.10 Daftar Objek 3D hasil modifikasi menggunakan Blender

No.	Nama Objek 3D	Gambar
1.	Bangun Ruang Persegi	
2.	Bangun Ruang Lingkaran	

3.2.5. Arsitektur Umum Sistem

Arsitektur sistem pada aplikasi permainan untuk anak penderita autisme menggunakan teknologi LMC ini didukung oleh dua perangkat yaitu komputer dan tentu saja LMC. Implementasi aplikasi juga memanfaatkan salah satu aplikasi *game engine* yang sudah terkenal keunggulannya yaitu Unity. Untuk pembuatan model tiga dimensi menggunakan aplikasi 3D *modeling* Blender. Arsitektur secara umum aplikasi ini terlihat pada Gambar 3.2.



Gambar 3.2 Rancangan Sederhana Arsitektur Aplikasi

3.2.6. Rancangan Antarmuka Aplikasi

Rancangan antarmuka aplikasi diperlukan untuk memberikan gambaran umum kepada pengguna bagaimana sistem yang ada dalam aplikasi ini berinteraksi dengan pengguna. Selain itu, rancangan ini juga memberikan gambaran bagi pengguna apakah tampilan yang sudah disediakan oleh aplikasi mudah untuk dipahami dan digunakan, sehingga akan muncul kesan *user experience* yang baik dan mudah.

3.2.6.1. Rancangan Antarmuka Menu Utama

Halaman ini merupakan tampilan awal dari aplikasi saat aplikasi pertama dijalankan. Pada tampilan Gambar 3.3 terdapat empat menu yang dapat dipilih oleh pengguna, menu-menu ini

adalah mode-mode permainan yang ada pada permainan ini. Selain mode-mode permainan ada pula tombol EXIT untuk keluar dari aplikasi ini.



Gambar 3.3 Rancangan Antarmuka Menu Utama

3.2.6.2. Rancangan Antarmuka Permainan “Get The Ball”

Halaman ini merupakan tampilan yang berisi antarmuka yang digunakan pemain untuk bermain aplikasi dengan jenis permainan “Get The Ball”. Pada tampilan Gambar 3.4 terdapat tombol HOME yang dapat ditekan oleh pengguna untuk kembali ke antarmuka menu utama, selain itu ada pula tombol *refresh* yang akan mengembalikan posisi *gameobject* ketika *gameobject* keluar dari layar permainan, akan ditampilkan juga skor yang telah diperoleh oleh pengguna dan juga waktu lama permainan yang telah berjalan.



Gambar 3.4 Rancangan Antarmuka Permainan “Get The Ball”

3.2.6.3. Rancangan Antarmuka Permainan “Match The Shape”

Halaman ini merupakan tampilan yang berisi antarmuka yang digunakan pemain untuk bermain aplikasi dengan jenis permainan “Match The Shape”. Pada tampilan Gambar 3.5 tombol HOME yang dapat ditekan oleh pengguna untuk kembali ke antarmuka menu utama, selain itu ada pula tombol *refresh* yang akan mengembalikan posisi *gameobject* ketika *gameobject* keluar dari layar permainan, akan ditampilkan juga skor yang telah diperoleh oleh pengguna dan juga waktu lama permainan yang telah berjalan.

3.2.6.4. Rancangan Antarmuka Permainan Find The Matching Animals

Halaman ini merupakan tampilan yang berisi antarmuka yang digunakan pemain untuk bermain aplikasi dengan jenis permainan “Find The Matching Animals”. Pada tampilan Gambar 3.6 terdapat tombol HOME yang dapat ditekan oleh pengguna untuk kembali ke antarmuka menu utama, selain itu ada pula tombol *refresh* yang akan mengembalikan posisi *gameobject* ketika *gameobject* keluar

dari layar permainan, akan ditampilkan juga skor yang telah diperoleh oleh pengguna dan juga waktu lama permainan yang telah berjalan.



Gambar 3.5 Rancangan Antarmuka Permainan “Match The Shape”



Gambar 3.6 Rancangan Antarmuka Permainan “Find The Matching Animals”

3.2.6.5. Rancangan Antarmuka Permainan “Pick The Fruit”

Halaman ini merupakan tampilan yang berisi antarmuka yang digunakan pemain untuk bermain aplikasi dengan jenis permainan “Pick The Fruit”. Pada tampilan Gambar 3.7 terdapat tombol HOME yang dapat ditekan oleh pengguna untuk kembali ke antarmuka menu utama, selain itu ada pula tombol *refresh* yang akan mengembalikan posisi *gameobject* ketika *gameobject* keluar dari layar permainan, akan ditampilkan juga skor yang telah diperoleh oleh pengguna dan juga waktu lama permainan yang telah berjalan.



Gambar 3.7 Rancangan Antarmuka Permainan “Pick The Fruit”

3.2.6.6. Rancangan Antarmuka Permainan Selesai

Halaman ini merupakan tampilan yang berisi antarmuka ketika pengguna telah selesai bermain. Pada tampilan Gambar 3.8 terdapat tombol HOME yang dapat ditekan oleh pengguna untuk kembali ke antarmuka menu utama jika ingin keluar dari aplikasi atau ingin memainkan permainan yang lain, selain itu ada juga tombol PLAYAGAIN untuk memainkan kembali permainan yang

telah dimainkan, selain itu ada pula tombol *refresh* yang akan mengembalikan posisi *gameobject* ketika *gameobject* keluar dari layar permainan, akan ditampilkan juga skor yang telah diperoleh oleh pengguna dan juga waktu lama permainan yang telah berjalan.



Gambar 3.8 Rancangan Antarmuka Permainan Selesai

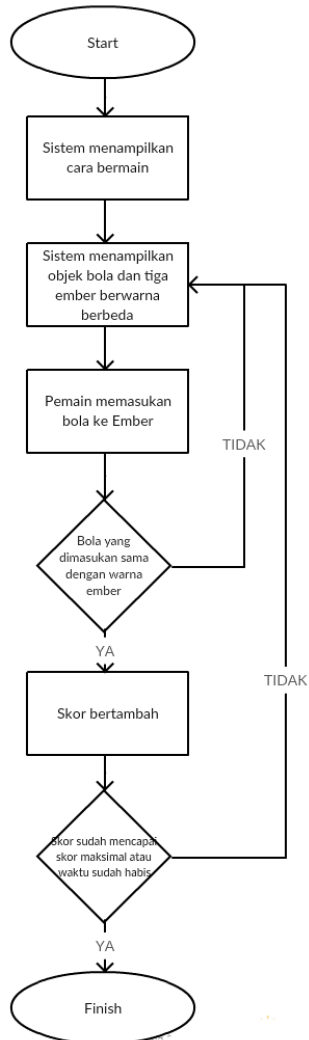
3.2.7. Perancangan Proses

Pada subbab ini akan dijelaskan mengenai rancangan proses yang dilakukan untuk mendukung skenario dan fungsionalitas yang sudah dirancang pada aplikasi ini. Proses-proses tersebut yang akan dibahas pada bab perancangan proses ini adalah yang pertama proses memainkan permainan “Get The Ball”. Proses kedua adalah proses memainkan permainan “Match The Shape”, setelah itu adalah proses memainkan permainan “Find The Matching Animals” dan yang terakhir adalah proses memainkan permainan “Pick The Fruit”.

3.2.7.1. Rancangan Proses Memainkan Permainan “Get The Ball”

Sebelum melakukan proses ini pengguna harus memastikan terlebih dahulu LMC sudah terhubung dengan komputer. Setelah bermain sudah menekan tombol permainan “Get The Ball”, objek-objek permainan akan muncul. Namun sebelumnya, pengguna akan ditampilkan oleh panduan bermain dari permainan terlebih dahulu sebelum dapat bermain. Setelah pemain sudah selesai membaca panduan permainan yang ditampilkan, panduan bermain akan hilang dan pengguna akan memulai permainan.

Pengguna akan terus memainkan permainan sampai tujuan dari permainan sudah tercapai yaitu memasukkan objek bola ke ember yang memiliki warna yang sama dengan warna bola. Setelah berhasil menyelesaikan tujuan dari permainan dengan benar, pengguna akan memperoleh satu skor. Permainan akan berakhir ketika skor maksimal telah tercapai atau waktu maksimal yang diberikan untuk bermain sudah terlewati. Setelah itu skor dan waktu akhir akan ditampilkan di layar dan pengguna dapat memilih apakah ingin memainkan permainan kembali atau ingin kembali ke *homescreen* dan memilih permainan lain. *Flowchart* proses permainan “Get The Ball” dapat dilihat pada Gambar 3.9.

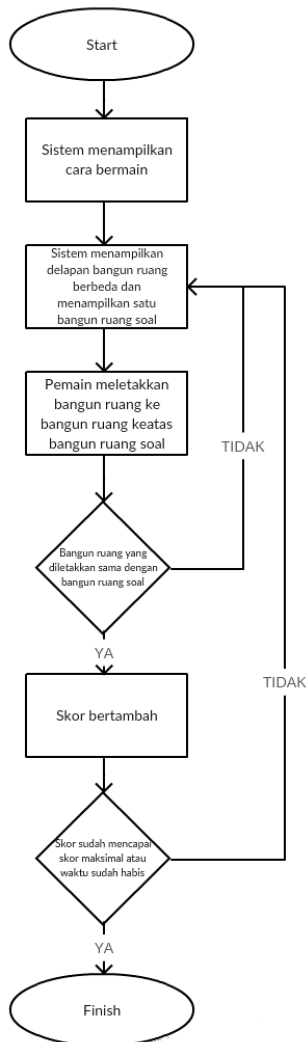


Gambar 3.9 *Flowchart* Proses Permainan “Get The Ball”

3.2.7.2. Rancangan Proses Memainkan Permainan “Match The Shape”

Sebelum melakukan proses ini pengguna harus memastikan terlebih dahulu LMC sudah terhubung dengan komputer. Setelah pemain sudah menekan tombol permainan “Match The Shape”, objek-objek permainan akan muncul. Namun sebelumnya, pengguna akan ditampilkan oleh panduan bermain dari permainan terlebih dahulu sebelum dapat bermain. Setelah pemain sudah selesai membaca panduan permainan yang ditampilkan, panduan bermain akan hilang dan pengguna akan memulai permainan.

Pengguna akan terus memainkan permainan sampai tujuan dari permainan sudah tercapai yaitu mengambil bangun dasar yang sama dengan bangun dasar yang sudah ditentukan oleh sistem lalu meletakkan bangun dasar tersebut ke atas bentuk bangun dasar yang ditampilkan oleh sistem. Setelah berhasil menyelesaikan tujuan dari permainan dengan benar, pengguna akan memperoleh satu skor. Permainan akan berakhir ketika skor maksimal telah tercapai atau waktu maksimal yang diberikan untuk bermain sudah terlewati. Setelah itu skor dan waktu akhir akan ditampilkan di layar dan pengguna dapat memilih apakah ingin memainkan permainan kembali atau ingin kembali ke *homescreen* dan memilih permainan lain. *Flowchart* proses permainan “Match The Shape” dapat dilihat pada Gambar 3.10.

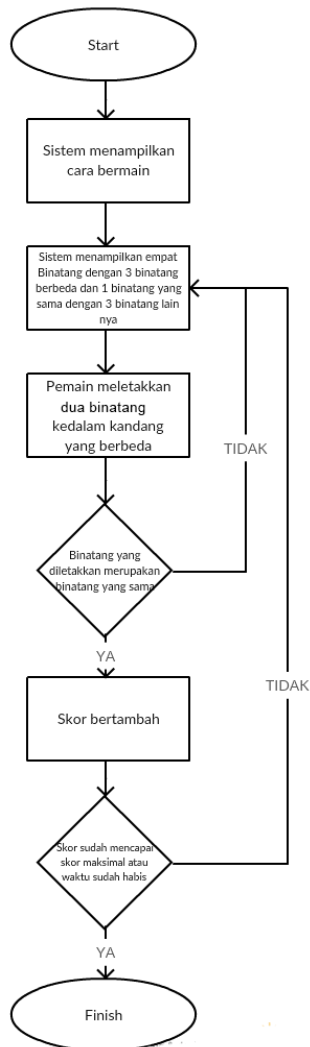


Gambar 3.10 Flowchart Proses Permainan “Match The Shape”

3.2.7.3. Rancangan Proses Memainkan Permainan “Find The Matching Animals”

Sebelum melakukan proses ini pengguna harus memastikan terlebih dahulu LMC sudah terhubung dengan komputer. Setelah pemain sudah menekan tombol permainan “Find The Matching Animals”, objek-objek permainan akan muncul. Namun sebelumnya, pengguna akan ditampilkan oleh panduan bermain dari permainan terlebih dahulu sebelum dapat bermain. Setelah pemain sudah selesai membaca panduan permainan yang ditampilkan, panduan bermain akan hilang dan pengguna akan memulai permainan.

Pengguna akan terus memainkan permainan sampai tujuan dari permainan sudah tercapai yaitu memindahkan dua ekor model binatang yang sama dari empat ekor model binatang yang ditampilkan ke dua kandang yang telah disediakan. Setelah berhasil menyelesaikan tujuan dari permainan dengan benar, pengguna akan memperoleh satu skor. Permainan akan berakhir ketika skor maksimal telah tercapai atau waktu maksimal yang diberikan untuk bermain sudah terlewati. Setelah itu skor dan waktu akhir akan ditampilkan di layar dan pengguna dapat memilih apakah ingin memainkan permainan kembali atau ingin kembali ke *homescreen* dan memilih permainan lain. *Flowchart* proses permainan “Find The Matching Animals” dapat dilihat pada Gambar 3.11.

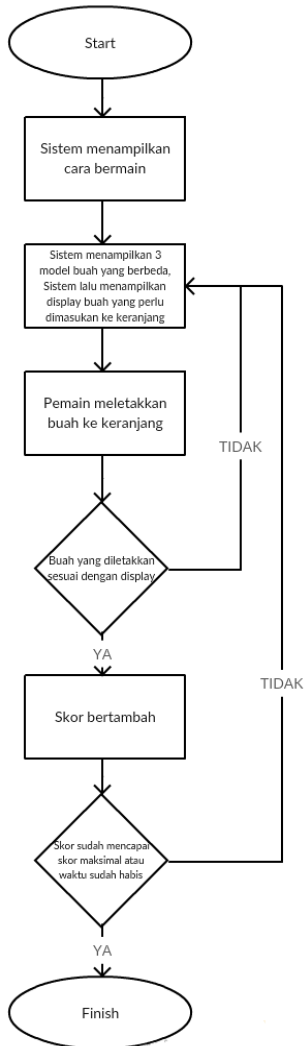


Gambar 3.11 Flowchart Proses Permainan “Find The Matching Animals”

3.2.7.4. Rancangan Proses Memainkan Permainan “Pick The Fruit”

Sebelum melakukan proses ini pengguna harus memastikan terlebih dahulu LMC sudah terhubung dengan komputer. Setelah bermain sudah menekan tombol permainan “Pick The Fruit”, objek-objek permainan akan muncul. Namun sebelumnya, pengguna akan ditampilkan oleh panduan bermain dari permainan terlebih dahulu sebelum dapat bermain. Setelah pemain sudah selesai membaca panduan permainan yang ditampilkan, panduan bermain akan hilang dan pengguna akan memulai permainan.

Pengguna akan terus memainkan permainan sampai tujuan dari permainan sudah tercapai yaitu tujuan dari permainan meletakkan bentuk buah yang berbentuk sama dari bentuk buah yang telah ditunjukkan oleh sistem ke keranjang yang telah disediakan. Setelah berhasil menyelesaikan tujuan dari permainan dengan benar, pengguna akan memperoleh satu skor. Permainan akan berakhir ketika skor maksimal telah tercapai atau waktu maksimal yang diberikan untuk bermain sudah terlewati. Setelah itu skor dan waktu akhir akan ditampilkan di layar dan pengguna dapat memilih apakah ingin memainkan permainan kembali atau ingin kembali ke *homescreen* dan memilih permainan lain. *Flowchart* proses permainan “Pick The Fruit” dapat dilihat pada Gambar 3.12.



Gambar 3.12 Flowchart Proses Permainan “Pick The Fruit”

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan perangkat lunak. Cakupan implementasi dari perancangan perangkat lunak tersebut meliputi proses penerapan dan pengimplementasian algoritme dan antarmuka.

4.1. *Lingkungan Implementasi*

Lingkungan implementasi Tugas Akhir dijelaskan pada Tabel 4.1.

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak

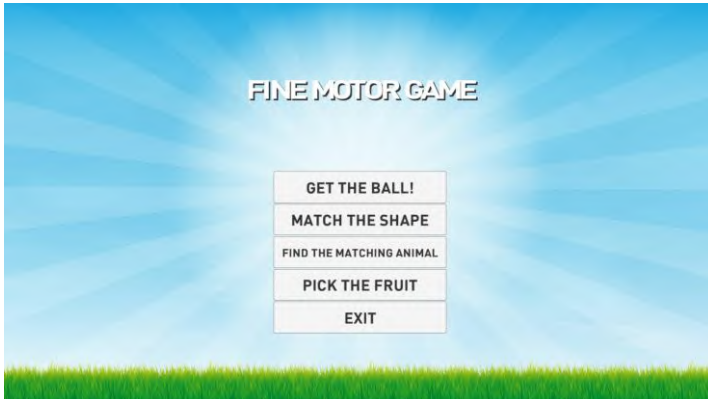
Perangkat Keras	Prosesor : - Intel(R) Core(TM) i5-3317U CPU @ 1.70GHz Memori : - 4 GB
Perangkat Lunak	Sistem Operasi : - Microsoft Windows 10 64-bit Perangkat Pengembang : - Blender 2.69 - Unity 5.2.3 - Visual Studio Community

4.2. *Implementasi Antarmuka*

Subbab ini membahas mengenai implementasi antarmuka pengguna yang meliputi implementasi antarmuka halaman menu utama, antarmuka halaman permainan “Get The Ball”, antarmuka halaman permainan “Match The Shape”, antarmuka halaman permainan “Find The Matching Animals”, antarmuka halaman permainan “Pick The Fruit”, dan antarmuka halaman *GameOver*.

4.2.1. Implementasi Antarmuka Halaman Menu Utama

Antarmuka halaman menu utama diimplementasikan dalam *scene* MainMenu.unity. *Scene* ini menggunakan bahasa C#. Tampilan halaman menu utama dijelaskan pada Gambar 4.1.



Gambar 4.1 Implementasi Tampilan Halaman Menu Utama

Tampilan halaman ini terdiri dari tombol menu yang berisi dari jenis-jenis permainan seperti “Get The Ball”, “Match The Shape”, “Find The Matching Shape” dan “Pick The Fruit”. Selain itu ada pula tombol EXIT. Tombol Jenis-jenis permainan ini digunakan untuk memulai permainan sesuai dengan jenis permainan yang tertulis pada tombol, sedangkan tombol EXIT digunakan untuk keluar dari aplikasi. Implementasi halaman menu utama dapat dijelaskan pada Kode Sumber 4.1.

```
public void gtbStart()
{
    SceneManager.LoadScene(1);
}

public void mtsStart()
{
    SceneManager.LoadScene(2);
}
```

```

public void ftmaStart()
{
    SceneManager.LoadScene(3);
}

public void ptfStart()
{
    SceneManager.LoadScene(4);
}

public void ExitPress()
{
    ExitMenu.enabled = true;
    Exit.enabled = false;
}

public void NoPress()
{
    ExitMenu.enabled = false;
    Exit.enabled = true;
}

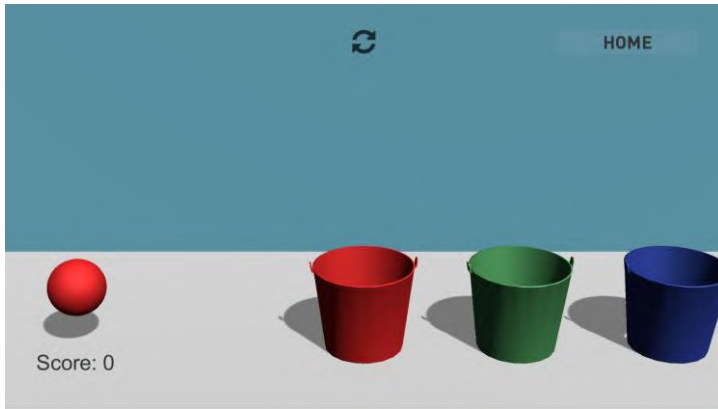
public void CloseApp()
{
    Application.Quit();
}

```

Kode Sumber 4.1 Implementasi Tampilan Halaman Menu Utama

4.2.2. Implementasi Antarmuka Halaman Permainan “Get The Ball”

Antarmuka halaman jenis permainan diimplementasikan dalam *scene* GetTheBall.unity. *Scene* ini menggunakan bahasa C#. Tampilan halaman jenis permainan dijelaskan pada Gambar 4.2.



Gambar 4.2 Implementasi Tampilan Halaman Permainan “Get The Ball”

Tampilan halaman ini terdiri dari tombol HOME. Tombol HOME digunakan untuk kembali ke halaman menu utama, selain tombol HOME ada pula jumlah skor sementara yang diperoleh oleh pengguna, selain itu ada pula tombol *refresh* yang akan mengembalikan posisi bola ketika bola keluar dari layar permainan. Implementasi halaman jenis permainan dapat dijelaskan pada Kode Sumber 4.2.

```
public void HomePress()
{
    homeMenu.enabled = true;
    hmButton.enabled = false;
}

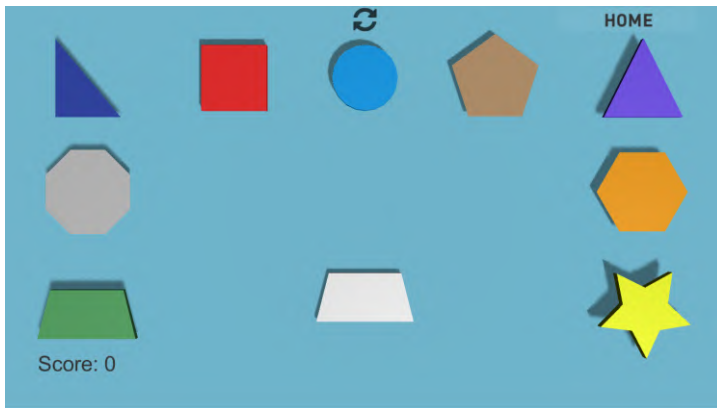
public void NoPress()
{
    homeMenu.enabled = false;
    hmButton.enabled = true;
}

public void GoHome()
{
    SceneManager.LoadScene(0);
}
```

Kode Sumber 4.2 Implementasi Antarmuka Halaman Permainan “Get The Ball”

4.2.3. Implementasi Antarmuka Halaman Permainan “Match The Shape”

Antarmuka halaman jenis permainan diimplementasikan dalam *scene* MatchTheShape.unity. *Scene* ini menggunakan bahasa C#. Tampilan halaman jenis permainan dijelaskan pada Gambar 4.3.



Gambar 4.3 Implementasi Tampilan Halaman Permainan “Match The Shape”

Tampilan halaman ini terdiri dari tombol HOME. Tombol HOME digunakan untuk kembali ke halaman menu utama, selain tombol HOME ada pula jumlah skor sementara yang diperoleh oleh pengguna, selain itu ada pula tombol *refresh* yang akan mengembalikan posisi bangun datar ketika bangun datar keluar dari layar permainan. Implementasi halaman jenis permainan dapat dijelaskan pada Kode Sumber 4.3.

```
public void HomePress()
{
    homeMenu.enabled = true;
    hmButton.enabled = false;
}

public void NoPress()
```

```

{
    homeMenu.enabled = false;
    hmButton.enabled = true;
}

public void GoHome()
{
    SceneManager.LoadScene(0);
}

```

Kode Sumber 4.3 Implementasi Antarmuka Halaman Permainan “Match The Shape”

4.2.4. Implementasi Antarmuka Halaman Permainan “Find The Matching Animals”

Antarmuka halaman jenis permainan diimplementasikan dalam *scene* FindTheMatchingAnimals.unity. *Scene* ini menggunakan bahasa C#. Tampilan halaman jenis permainan dijelaskan pada Gambar 4.4.



Gambar 4.4 Implementasi Tampilan Halaman Permainan “Find The Matching Animals”

Tampilan halaman ini terdiri dari tombol HOME. Tombol HOME digunakan untuk kembali ke halaman menu utama, selain tombol HOME ada pula jumlah skor sementara yang diperoleh oleh pengguna, selain itu ada pula tombol *refresh* yang akan

mengembalikan posisi binatang ketika binatang keluar dari layar permainan. Implementasi halaman jenis permainan dapat dijelaskan pada Kode Sumber 4.4.

```

public void HomePress()
{
    homeMenu.enabled = true;
    hmButton.enabled = false;
}

public void NoPress()
{
    homeMenu.enabled = false;
    hmButton.enabled = true;
}

public void GoHome()
{
    SceneManager.LoadScene(0);
}

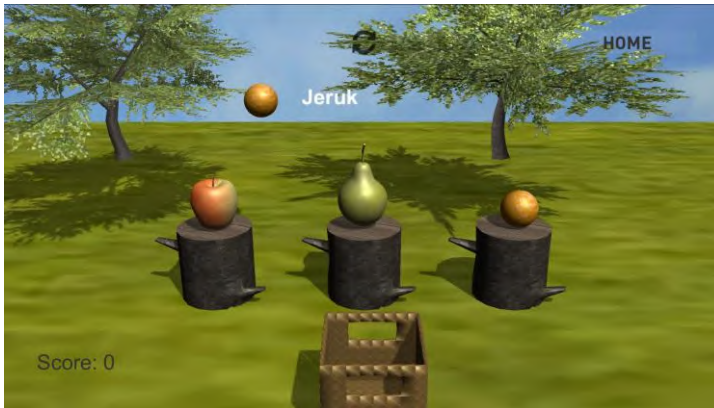
```

**Kode Sumber 4.4 Implementasi Antarmuka Halaman Permainan
“Find The Matching Animals”**

4.2.5. Implementasi Antarmuka Halaman Permainan “Pick The Fruit”

Antarmuka halaman jenis permainan diimplementasikan dalam *scene* PickTheFruit.unity. *Scene* ini menggunakan bahasa C#. Tampilan halaman jenis permainan dijelaskan pada Gambar 4.5.

Tampilan halaman ini terdiri dari tombol HOME. Tombol HOME digunakan untuk kembali ke halaman menu utama, selain tombol HOME ada pula jumlah skor sementara yang diperoleh oleh pengguna, selain itu ada pula tombol *refresh* yang akan mengembalikan posisi buah ketika buah keluar dari layar permainan. Implementasi halaman jenis permainan dapat dijelaskan pada Kode Sumber 4.5.



Gambar 4.5 Implementasi Tampilan Halaman Permainan “Pick The Fruit”

```
//menmunculkan menu ketika tombol home ditekan
public void HomePress()
{
    homeMenu.enabled = true;
    hmButton.enabled = false;
}

//mengembalikan menu ketika tombol no ditekan
public void NoPress()
{
    homeMenu.enabled = false;
    hmButton.enabled = true;
}

//Kembali ke scene 0 yaitu scene MainMenu
public void GoHome()
{
    SceneManager.LoadScene(0);
}
```

Kode Sumber 4.5 Implementasi Antarmuka Halaman Permainan “Pick The Fruit”

4.2.6. Implementasi Antarmuka Halaman Ketika Permainan Selesai

Antarmuka halaman GameOver diimplementasikan dalam kode berbahasa C#. Halaman GameOver muncul dalam setiap

permainan saat pemain kehabisan waktu bermain atau telah mencapai skor tertinggi yang telah ditetapkan. Dalam halaman *GameOver*, terdapat skor yang telah dicapai pemain, waktu penyelesaian permainan, tombol HOME dan Tombol PLAY AGAIN. Tampilan halaman Challenge saat permainan dimulai adalah seperti pada Gambar 4.6.

Tampilan halaman ini terdiri dari skor yang telah dicapai pemain, tombol HOME dan Tombol PLAY AGAIN. Skor yang ditampilkan adalah skor akhir yang telah dicapai pemain, tombol *Back* berguna untuk kembali ke halaman menu utama, tombol *Replay* berguna untuk mengulang permainan yang sama. Implementasi halaman *GameOver* dapat dijelaskan pada Kode Sumber 4.6.



Gambar 4.6 Implementasi Tampilan Halaman ketika permainan selesai

```
public void HomePress()
{
    SceneManager.LoadScene(0);
}

public void PlayAgain()
{
```

```

        Scene scene = SceneManager.GetActiveScene();
        SceneManager.LoadScene(scene.name);
    }

    // Update is called once per frame
    void Update()
    {
        //memulai merekam waktu permainan
        if (record)
            currentTime += 1 * Time.deltaTime;

        //memunculkan menu ketika skor sudah mencapai skor
maksimal
        if (score >= scoreMax)
        {
            record = false;
            compTime.text = currentTime.ToString();
            Time.timeScale = 0;
            endMenu.enabled = true;
            finalScore.text = score.ToString();
            Cursor.visible = true;
        }
    }
}

```

Kode Sumber 4.6 Implementasi Antarmuka Halaman ketika permainan selesai

4.3. Implementasi Proses

Subbab ini akan menjelaskan tentang implementasi dari rancangan proses yang dijelaskan pada Bab III. Penjelasan mengenai implementasi proses ini dibagi berdasarkan komponen-komponen aplikasi. Berikut ini merupakan penjelasan dari tiap-tiap komponen.

4.3.1. Proses Memainkan Permainan “Get The Ball”

Proses ini merupakan proses dari permainan “Get The Ball” yang dapat dilakukan oleh pengguna. Dalam permainan ini pengguna akan memasukkan objek bola ke ember yang memiliki warna yang sama dengan warna bola. Permainan ini dilakukan dengan menggerakkan tangan kanan di atas Leap Motion Controller (LMC) dan mengepalkan tangan untuk mengambil objek yang ingin diambil. Setelah berhasil menyelesaikan tujuan

dari permainan dengan benar, pengguna akan memperoleh satu skor. Permainan akan berakhir ketika skor maksimal telah tercapai atau waktu maksimal yang diberikan untuk bermain sudah terlewati. Setelah itu skor dan waktu akhir akan ditampilkan di layar dan pengguna dapat memilih apakah ingin memainkan permainan kembali atau ingin kembali ke *homescreen* dan memilih permainan lain.

Implementasi dari proses ini dengan menggunakan fungsi yang sudah disediakan oleh Unity yaitu *OnCollisionEnter*. Fungsi *OnCollisionEnter* merupakan fungsi pendeteksian benturan atau tabrakan antar objek. Objek yang dimaksud pada fungsi ini adalah benturan bola dan ember. Apabila terjadi benturan maka aplikasi akan mengecek apakah warna dari bola yang dimasukkan sesuai dengan warna ember, jika benar pengguna akan mendapatkan skor jika tidak, bola baru akan di munculkan. Potongan kode untuk memainkan permainan “Get The Ball” dijelaskan pada Kode Sumber 4.7.

```
void OnCollisionEnter(Collision bucket)
{
    if (bucket.transform.tag == "bucketR")
    {
        if (rand == 1)
        {
            score++;
            SetUI();
            Respawn();
        }
        else
            Respawn();
    }

    if (bucket.transform.tag == "bucketG")
    {
        if (rand == 2)
        {
            score++;
            SetUI();
            Respawn();
        }
        else
    }
```

```

        Respawn();
    }
    if (bucket.transform.tag == "bucketB")
    {
        if (rand == 3)
        {
            score++;
            SetUI();
            Respawn();
        }
        else
            Respawn();
    }
}

void Respawn()
{
    transform.position = spawn;
    GetComponent<Rigidbody>().velocity = Vector3.zero;
    GetComponent<Rigidbody>().angularVelocity = Vector3.zero;
    rand = Random.Range(1, 4);
    if (rand == 1)
        gameObject.GetComponent<Renderer>().material.color =
red;
    else if (rand == 2)
        gameObject.GetComponent<Renderer>().material.color =
green;
    else if (rand == 3)
        gameObject.GetComponent<Renderer>().material.color =
blue;

    Debug.Log(rand);
}

void SetUI()
{
    scoreUI.text = "Score: " + score.ToString();
}

```

Kode Sumber 4.7 Memainkan Permainan “Get The Ball”

4.3.2. Proses Memainkan Permainan “Match The Shape”

Proses ini merupakan proses dari permainan “Match The Shape” yang dapat dilakukan oleh pengguna. Dalam permainan ini

pengguna akan mengambil bangun dasar yang sama dengan bangun dasar yang sudah ditentukan oleh sistem lalu meletakkan bangun dasar tersebut ke atas bentuk bangun dasar yang ditampilkan oleh sistem. Permainan ini dilakukan dengan menggerakkan tangan kanan di atas LMC dan mengepalkan tangan untuk mengambil objek yang ingin diambil. Setelah berhasil menyelesaikan tujuan dari permainan dengan benar, pengguna akan memperoleh satu skor. Permainan akan berakhir ketika skor maksimal telah tercapai atau waktu maksimal yang diberikan untuk bermain sudah terlewati. Setelah itu skor dan waktu akhir akan ditampilkan di layar dan pengguna dapat memilih apakah ingin memainkan permainan kembali atau ingin kembali ke *homescreen* dan memilih permainan lain.

Implementasi dari proses ini dengan menggunakan fungsi yang sudah disediakan oleh Unity yaitu *OnTriggerEnter*. Fungsi *OnTriggerEnter* merupakan fungsi pendeteksian benturan atau tabrakan antar objek. Objek yang dimaksud pada fungsi ini adalah benturan bangun datar dan bangun datar yang ditampilkan sistem. Apabila terjadi benturan maka aplikasi akan mengecek apakah bangun datar yang diletakkan sesuai dengan bangun datar yang ditampilkan sistem, jika benar pengguna akan mendapatkan skor jika tidak, bangun datar baru akan di munculkan. Potongan kode untuk memainkan permainan “Match The Shape” dijelaskan pada Kode Sumber 4.8.

```
//mengecek apakah yang masuk ke trigger memiliki tag sesuai dengan
bangun datar yang dimunculkan
IEnumerator OnTriggerEnter(Collider col)
{
    collided = true;

    if (col.transform.tag == "square")
    {
        if (collided && MagneticPinchMod.grabbed_ == null &&
random == 1)
        {
            //serangkaian kegiatan yang dilakukan,
menghilangkan display bangun datar yang ada, memunculkan display
baru dan mengembalikan posisi bangun datar yang berubah
```

```

        yield return new WaitForSeconds(3);
        Destroy(display);
        RestorePosition();
        GenerateDisplay();
        Debug.Log(col.transform.name);
    }
    else if (collided && MagneticPinchMod.grabbed_ == null
&& random != 1)
    {
        yield return new WaitForSeconds(3);
        Destroy(display);
        RestorePosition();
        GenerateDisplay();
        Debug.Log(col.transform.name);
    }
}

else if (col.transform.tag == "triangle1")
{
    if (collided && MagneticPinchMod.grabbed_ == null &&
random == 2)
    {
        yield return new WaitForSeconds(3);
        Destroy(display);
        RestorePosition();
        GenerateDisplay();
        ScoreCount();
        Debug.Log(col.transform.name);
    }
    else if (collided && MagneticPinchMod.grabbed_ == null
&& random != 2)
    {
        yield return new WaitForSeconds(3);
        Destroy(display);
        RestorePosition();
        GenerateDisplay();
        Debug.Log(col.transform.name);
    }
}

else if (col.transform.tag == "triangle2")
{
    if (collided && MagneticPinchMod.grabbed_ == null &&
random == 3)
    {
        yield return new WaitForSeconds(3);
        Destroy(display);
        RestorePosition();
    }
}

```



```

        GenerateDisplay();
        ScoreCount();
        Debug.Log(col.transform.name);
    }
    else if (collided && MagneticPinchMod.grabbed_ == null
&& random != 3)
    {
        yield return new WaitForSeconds(3);
        Destroy(display);
        RestorePosition();
        GenerateDisplay();
        Debug.Log(col.transform.name);
    }
}

    else if (col.transform.tag == "trapezium")
    {
        if (collided && MagneticPinchMod.grabbed_ == null &&
random == 4)
        {
            yield return new WaitForSeconds(3);
            Destroy(display);
            RestorePosition();
            GenerateDisplay();
            ScoreCount();
            Debug.Log(col.transform.name);
        }
        else if (collided && MagneticPinchMod.grabbed_ == null
&& random != 4)
        {
            yield return new WaitForSeconds(3);
            Destroy(display);
            RestorePosition();
            GenerateDisplay();
            Debug.Log(col.transform.name);
        }
    }

    else if (col.transform.tag == "circle")
    {
        if (collided && MagneticPinchMod.grabbed_ == null &&
random == 5)
        {
            yield return new WaitForSeconds(3);
            Destroy(display);
            RestorePosition();
            GenerateDisplay();
            ScoreCount();
        }
    }
}

```

```

        Debug.Log(col.transform.name);
    }
    else if (collided && MagneticPinchMod.grabbed_ == null
&& random != 5)
    {
        yield return new WaitForSeconds(3);
        Destroy(display);
        RestorePosition();
        GenerateDisplay();
        Debug.Log(col.transform.name);
    }
}
else if (col.transform.tag == "star")
{
    if (collided && MagneticPinchMod.grabbed_ == null &&
random == 6)
    {
        yield return new WaitForSeconds(3);
        Destroy(display);
        RestorePosition();
        GenerateDisplay();
        ScoreCount();
        Debug.Log(col.transform.name);
    }
    else if (collided && MagneticPinchMod.grabbed_ == null
&& random != 6)
    {
        yield return new WaitForSeconds(3);
        Destroy(display);
        RestorePosition();
        GenerateDisplay();
        Debug.Log(col.transform.name);
    }
}
else if (col.transform.tag == "pentagon")
{
    if (collided && MagneticPinchMod.grabbed_ == null &&
random == 7)
    {
        yield return new WaitForSeconds(3);
        Destroy(display);
        RestorePosition();
        GenerateDisplay();
        ScoreCount();
        Debug.Log(col.transform.name);
    }
    else if (collided && MagneticPinchMod.grabbed_ == null
&& random != 7)

```

```

        {
            yield return new WaitForSeconds(3);
            Destroy(display);
            RestorePosition();
            GenerateDisplay();
            Debug.Log(col.transform.name);
        }
    }
    else if (col.transform.tag == "octagon")
    {
        if (collided && MagneticPinchMod.grabbed_ == null &&
random == 8)
        {
            yield return new WaitForSeconds(3);
            Destroy(display);
            RestorePosition();
            GenerateDisplay();
            ScoreCount();
            Debug.Log(col.transform.name);
        }
        else if (collided && MagneticPinchMod.grabbed_ == null
&& random != 8)
        {
            yield return new WaitForSeconds(3);
            Destroy(display);
            RestorePosition();
            GenerateDisplay();
            Debug.Log(col.transform.name);
        }
    }
    else if (col.transform.tag == "hexagon")
    {
        if (collided && MagneticPinchMod.grabbed_ == null &&
random == 9)
        {
            yield return new WaitForSeconds(3);
            Destroy(display);
            RestorePosition();
            GenerateDisplay();
            ScoreCount();
            Debug.Log(col.transform.name);
        }
        else if (collided && MagneticPinchMod.grabbed_ == null
&& random != 9)
        {
            yield return new WaitForSeconds(3);
            Destroy(display);
            RestorePosition();

```

```

        GenerateDisplay();
        Debug.Log(col.transform.name);
    }
}

void OnTriggerExit()
{
    collided = false;
}

```

Kode Sumber 4.8 Memainkan Permainan Match The Shape

4.3.3. Proses Memainkan Permainan “Find The Matching Animals”

Proses ini merupakan proses dari permainan “Find The Matching Animals” yang dapat dilakukan oleh pengguna. Dalam permainan ini pengguna akan memindahkan dua ekor model binatang yang sama dari empat ekor model binatang yang ditampilkan ke dua kandang yang telah disediakan. Permainan ini dilakukan dengan menggerakkan tangan kanan di atas LMC dan mengepalkan tangan untuk mengambil objek yang ingin diambil. Setelah berhasil menyelesaikan tujuan dari permainan dengan benar, pengguna akan memperoleh satu skor. Permainan akan berakhir ketika skor maksimal telah tercapai atau waktu maksimal yang diberikan untuk bermain sudah terlewati. Setelah itu skor dan waktu akhir akan ditampilkan di layar dan pengguna dapat memilih apakah ingin memainkan permainan kembali atau ingin kembali ke *homescreen* dan memilih permainan lain.

Implementasi dari proses ini dengan menggunakan fungsi yang sudah disediakan oleh Unity yaitu `OnTriggerEnter`. Fungsi `OnTriggerEnter` merupakan fungsi pendeteksian benturan atau tabrakan antar objek. Objek yang dimaksud pada fungsi ini adalah benturan binatang dan collider yang ada pada kandang. Apabila terjadi benturan maka aplikasi akan mengecek apakah binatang yang ada di kandang pertama dan binatang yang ada di kandang

kedua sama, jika benar pengguna akan mendapatkan skor jika tidak, empat ekor binatang baru akan ditampilkan. Potongan kode untuk memainkan permainan “Find The Matching Animals” dijelaskan pada Kode Sumber 4.9.

```
void OnTriggerEnter(Collider col)
{
    if (CheckCollider.animal == col.transform.tag)
    {
        DestroyClone();
        DisplayAnimal();
        ScoreCount();
    }
    else
    {
        DestroyClone();
        DisplayAnimal();
    }
}

void DisplayAnimal()
{
    animalSet = GenerateRandom();
    for (int i = 0; i < animalSet.Count; i++)
    {
        print(animalSet[i]);
    }

    int num = 4;
    while (num > 0)
    {
        tagged = AnimalTag(animalSet);
        ori = GameObject.FindGameObjectWithTag(tagged);
        if (animalSet.Count == 3)
        {
            clone = Instantiate(ori, aPos,
Quaternion.identity) as GameObject;
            clone.name = "Clone 1";
            cloneList.Add(clone.transform.name);
            num--;
        }
        else if (animalSet.Count == 2)
        {
            clone = Instantiate(ori, bPos,
Quaternion.identity) as GameObject;
```

```

        clone.name = "Clone 2";
        cloneList.Add(clone.transform.name);
        num--;
    }
    else if (animalSet.Count == 1)
    {
        clone = Instantiate(ori, cPos,
Quaternion.identity) as GameObject;
        clone.name = "Clone 3";
        cloneList.Add(clone.transform.name);
        num--;
    }
    else if (animalSet.Count == 0)
    {
        clone = Instantiate(ori, dPos,
Quaternion.identity) as GameObject;
        clone.name = "Clone 4";
        cloneList.Add(clone.transform.name);
        break;
    }
}

}

void DestroyClone()
{
    for (int i = 0; i < cloneList.Count; i++)
    {
        Destroy(GameObject.Find(cloneList[i]));
    }
}

string AnimalTag(List<int> arr)
{
    if (arr[0] == 1 && animalSet.Count != 0)
    {
        animalSet.RemoveAt(0);
        return ("horse");
    }
    else if (arr[0] == 2 && animalSet.Count != 0)
    {
        animalSet.RemoveAt(0);
        return ("rabbit");
    }
    else if (arr[0] == 3 && animalSet.Count != 0)
    {
        animalSet.RemoveAt(0);
        return ("tiger");
    }
}

```

```

    }
    else if (arr[0] == 4 && animalSet.Count != 0)
    {
        animalSet.RemoveAt(0);
        return ("elephant");
    }
    else
        return ("");
}

List<int> GenerateRandom()
{
    for (int i = 0; i < randSet.Length - 1; i++)
    {
        int num = randSet[Random.Range(0, randSet.Length)];
        if (dis.Contains(num) != true)
            dis.Add(num);
        else
            i--;
    }

    int r = Random.Range(1, dis.Count);
    dis.Add(dis[r]);

    for (int i = 0; i < dis.Count; i++)
    {
        int temp = dis[i];
        int randomIndex = Random.Range(i, dis.Count);
        dis[i] = dis[randomIndex];
        dis[randomIndex] = temp;
    }

    return dis;
}

void ScoreCount()
{
    score++;
    scoreUI.text = "Score: " + score.ToString();
}

```

Kode Sumber 4.9 Memainkan Permainan “Find The Matching Animals”

4.3.4. Proses Memainkan Permainan “Pick The Fruit”

Proses ini merupakan proses dari permainan “Pick The Fruit” yang dapat dilakukan oleh pengguna. Dalam permainan ini pengguna akan meletakkan bentuk buah yang berbentuk sama dari bentuk buah yang telah ditunjukkan oleh sistem ke keranjang yang telah disediakan. Permainan ini dilakukan dengan menggerakkan tangan kanan di atas LMC dan mengepalkan tangan untuk mengambil objek yang ingin diambil. Setelah berhasil menyelesaikan tujuan dari permainan dengan benar, pengguna akan memperoleh satu skor. Permainan akan berakhir ketika skor maksimal telah tercapai atau waktu maksimal yang diberikan untuk bermain sudah terlewati. Setelah itu skor dan waktu akhir akan ditampilkan di layar dan pengguna dapat memilih apakah ingin memainkan permainan kembali atau ingin kembali ke *homescreen* dan memilih permainan lain.

Implementasi dari proses ini dengan menggunakan fungsi yang sudah disediakan oleh Unity yaitu *OnCollisionEnter*. Fungsi *OnCollisionEnter* merupakan fungsi pendeteksian benturan atau tabrakan antar objek. Objek yang dimaksud pada fungsi ini adalah benturan buah dan *collider* yang ada pada keranjang. Apabila terjadi benturan maka aplikasi akan mengecek apakah buah yang dimasukkan ke keranjang sesuai dengan buah yang ditampilkan oleh sistem, jika benar pengguna akan mendapatkan skor jika tidak, empat ekor binatang baru akan ditampilkan. Potongan kode untuk memainkan permainan “Pick The Fruit” dijelaskan pada Kode Sumber 4.9.

```
void OnCollisionEnter(Collision col)
{
    if (col.transform.tag == "apple")
    {
        if (random == 1)
        {
            DestroyDisplay();
            GenerateDisplay();
            ScoreCount();
            Respawn(col.gameObject.tag);
            Destroy(col.gameObject);
        }
    }
}
```



```

        RestorePosition();
    }
    else {
        DestroyDisplay();
        GenerateDisplay();
        Respawn(col.gameObject.tag);
        Destroy(col.gameObject);
        RestorePosition();
    }
}
else if (col.transform.tag == "pear")
{
    if (random == 2)
    {
        DestroyDisplay();
        GenerateDisplay();
        ScoreCount();
        Respawn(col.gameObject.tag);
        Destroy(col.gameObject);
        RestorePosition();
    }
    else {
        DestroyDisplay();
        GenerateDisplay();
        Respawn(col.gameObject.tag);
        Destroy(col.gameObject);
        RestorePosition();
    }
}
else if (col.transform.tag == "orange")
{
    if (random == 3)
    {
        DestroyDisplay();
        GenerateDisplay();
        ScoreCount();
        Respawn(col.gameObject.tag);
        Destroy(col.gameObject);
        RestorePosition();
    }
    else {
        DestroyDisplay();
        GenerateDisplay();
        Respawn(col.gameObject.tag);
        Destroy(col.gameObject);
        RestorePosition();
    }
}
}

```

```

    }

    void Respawn(string fName)
    {
        GameObject original =
GameObject.FindGameObjectWithTag(fName);
        if (fName == "apple")
        {
            clone = Instantiate(original, aPos,
Quaternion.identity) as GameObject;
            clone.name = "Apple";
        }
        else if (fName == "pear")
        {
            clone = Instantiate(original, bPos,
Quaternion.identity) as GameObject;
            clone.name = "Pear";
        }

        else if (fName == "orange")
        {
            clone = Instantiate(original, cPos,
Quaternion.identity) as GameObject;
            clone.name = "Orange";
        }

        GetComponent<Rigidbody>().velocity = Vector3.zero;
        GetComponent<Rigidbody>().angularVelocity = Vector3.zero;
    }

    void GenerateDisplay()
    {
        random = Random.Range(1, 4);
        Debug.Log(random);
        switch (random)
        {
            case 1:
                fruit = GameObject.FindGameObjectWithTag("apple");
                SetName("Apel");
                break;
            case 2:
                fruit = GameObject.FindGameObjectWithTag("pear");
                SetName("Pir");
                break;
            case 3:
                fruit =
GameObject.FindGameObjectWithTag("orange");

```

```

        SetName("Jeruk");
        break;
    }
    display = Instantiate(fruit, dPos, Quaternion.identity) as
GameObject;
    display.tag = "display";
    display.GetComponent<Rigidbody>().velocity = Vector3.zero;
    display.GetComponent<Rigidbody>().isKinematic = true;
}

void DestroyDisplay()
{
    Destroy(display);
}

void RestorePosition()
{
    GameObject.FindGameObjectWithTag("apple").transform.position =
aPos;

    GameObject.FindGameObjectWithTag("apple").transform.rotation =
Quaternion.identity;

    GameObject.FindGameObjectWithTag("apple").GetComponent<Rigidbody>()
).velocity = Vector3.zero;

    GameObject.FindGameObjectWithTag("pear").transform.position =
bPos;

    GameObject.FindGameObjectWithTag("pear").transform.rotation =
Quaternion.identity;

    GameObject.FindGameObjectWithTag("pear").GetComponent<Rigidbody>()
).velocity = Vector3.zero;

    GameObject.FindGameObjectWithTag("orange").transform.position =
cPos;

    GameObject.FindGameObjectWithTag("orange").transform.rotation =
Quaternion.identity;

    GameObject.FindGameObjectWithTag("orange").GetComponent<Rigidbody>
()).velocity = Vector3.zero;
}
void ScoreCount()
{
    score++;
}

```

```
        scoreUI.text = "Score: " + score.ToString();  
    }  
  
    void SetName(string fname)  
    {  
        nameUI.text = fname;  
    }
```

Kode Sumber 4.10 Memainkan Permainan “Pick The Fruit”

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada aplikasi yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsionalitas sistem dan kegunaan sistem. Pengujian fungsionalitas mengacu pada kasus penggunaan pada bab tiga. Pengujian kegunaan program dilakukan dengan mengetahui tanggapan dari pengguna terhadap sistem. Hasil evaluasi menjabarkan tentang rangkuman hasil pengujian pada bagian akhir bab ini.

5.1. Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan pada lingkungan dan alat kakas seperti yang tertera pada Tabel 5.1.

Tabel 5.1 Lingkungan Pengujian Sistem

Perangkat Keras	Prosesor : - Intel(R) Core(TM) i5-3317U CPU @ 1.70GHz Memori : - 4 GB
Perangkat Lunak	Sistem Operasi : - Microsoft Windows 10 64-bit Perangkat Pengembang : - Blender 2.69 - Unity 5.2.3 - Visual Studio Community

5.2. Pengujian Aplikasi

Pengujian aplikasi dilakukan untuk mengetahui kesesuaian keluaran dari tiap tahap atau langkah penggunaan fitur terhadap skenario yang dipersiapkan. Berikut ini penjabaran skenario dan

hasil uji coba yang dilakukan terhadap perangkat lunak yang dibangun.

5.2.1. Skenario Pengujian Fungsionalitas

Pada subbab ini dijelaskan beberapa skenario uji coba perangkat lunak secara mandiri berdasarkan metode kotak hitam sebagai dasar tolak ukur keberhasilan. Pengujian fungsionalitas yang terdapat pada aplikasi dijabarkan sebagai berikut:

- a. Uji coba memainkan permainan “Get The Ball”.
- b. Uji coba memainkan permainan “Match The Shape”.
- c. Uji coba memainkan permainan “Find The Matching Animals”.
- d. Uji coba memainkan permainan “Pick The Fruit”.

Daftar uji coba tersebut merupakan pengujian yang dilakukan untuk menguji setiap kasus penggunaan pada perangkat lunak yang dibangun. Dibuat beberapa skenario yang dilakukan pada setiap daftar pengujian tersebut. Penjelasan mengenai cara dan hasil pengujian fungsionalitas perangkat lunak dibahas pada subbab hasil uji coba.

5.2.2. Hasil Uji Coba Fungsionalitas Aplikasi

Pada subbab ini dijelaskan secara detail mengenai skenario yang dilakukan dan hasil yang didapatkan dari pengujian fungsionalitas perangkat lunak yang dibangun. Penjelasan disajikan dengan menampilkan kondisi awal, masukan, keluaran, hasil yang dicapai, dan kondisi akhir. Berikut ini merupakan penjabaran skenario dan hasil pengujian yang dicapai pada tiap-tiap fungsionalitas perangkat lunak.

Pengujian aplikasi dilakukan untuk mengetahui kesesuaian keluaran dari tiap tahap atau langkah keempat permainan yang ada pada aplikasi ini dan juga performa dari aplikasi ini, apakah aplikasi ini dapat menjalankan tiap permainan tanpa *error* yang dapat mengganggu permainan. Pengujian dibagi menjadi dua, yaitu:

1. Pengujian oleh penulis

Pada pengujian ini akan dilakukan oleh penulis meliputi pengujian kelancaran dari permainan “Get The Ball”, permainan “Match The Shape”, permainan “Find The Matching Animals”, dan permainan “Pick The Fruit”.

2. Pengujian oleh anak penderita autisme

Pada pengujian ini akan dilakukan oleh orang lain dengan menggunakan kuesioner untuk mengetahui kemudahan dalam memainkan permainan ini, kenyamanan dalam memakai aplikasi ini dan juga apakah aplikasi ini dapat dijadikan sebagai alternatif dari terapi okupasi yang sudah ada.

5.2.3. Pengujian oleh Penulis

Pada subbab ini dijelaskan uji coba perangkat lunak yang dilakukan oleh penulis sebagai dasar tolak ukur keberhasilan pada fungsionalitas aplikasi. Pengujian yang dilakukan pada aplikasi, yaitu pengujian memainkan permainan “Get The Ball”, permainan “Match The Shape”, permainan “Find The Matching Animals”, dan permainan “Pick The Fruit”. Penjelasan mengenai cara dan hasil pengujian perangkat lunak oleh penulis dibahas pada subbab berikut.

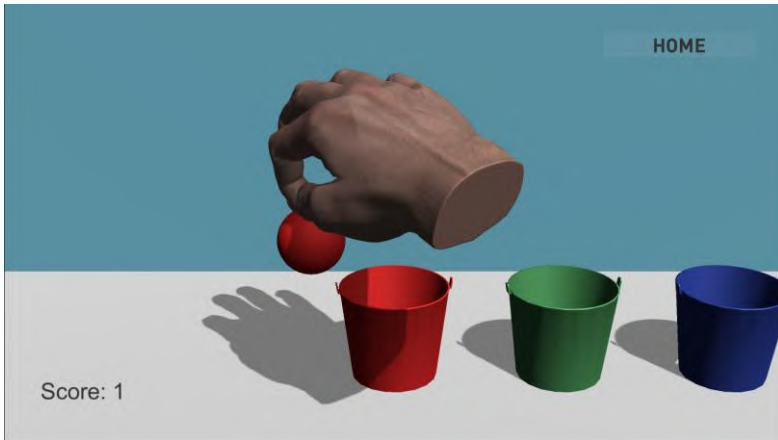
5.2.3.1. Uji Coba Memainkan Permainan “Get The Ball”

Pengujian dalam memainkan permainan “Get The Ball” ini dilakukan untuk menguji keberhasilan fungsionalitas utama dari permainan ini. Hasil dari pengujian ini akan secara lengkap dijabarkan pada Tabel 5.2. Hasil dari skenario uji coba tersebut dapat dilihat pada Gambar 5.1.

Tabel 5.2 Hasil Uji Coba Memainkan Permainan “Get The Ball”

ID	UJ- 001
Nama	Uji Coba Memainkan Permainan “Get The Ball”
Tujuan uji coba	Menguji fitur untuk memainkan permainan “Get The Ball”

ID	UJ- 001
Kondisi awal	Proses <i>rendering</i> objek sudah selesai
<i>Skenario 1</i>	<i>Pengguna mengambil bola dan memasukan nya ke ember yang memiliki warna yang sama</i>
Masukan	Memasukan bola ke ember
Keluaran yang diharapkan	Skor bertambah dan bola yang baru muncul
Hasil uji coba	Berhasil
Kondisi akhir	<ul style="list-style-type: none"> • Jika permainan sudah mencapai skor maksimal, pengguna akan dimunculkan layar <i>Game Over</i>. • Jika permainan belum mencapai skor maksimal, pengguna akan kembali berada di layar permainan “Get The Ball”.
<i>Skenario 2</i>	<i>Pengguna memasukkan bola ke ember yang tidak memiliki warna yang sama</i>
Masukan	Memasukkan bola ke ember
Keluaran yang diharapkan	Skor tetap dan bola yang baru muncul
Hasil uji coba	Berhasil
Kondisi akhir	Pengguna akan berada di layar permainan “Get The Ball”
<i>Skenario 3</i>	<i>Pengguna menekan tombol HOME pada Halaman permainan</i>
Masukan	Klik tombol HOME
Keluaran yang diharapkan	Pengguna akan dimunculkan menu untuk konfirmasi apakah benar akan kembali ke <i>homescreen</i> atau tidak. Jika pengguna menekan <i>YES</i> , pengguna akan kembali ke <i>main menu</i> , jika pengguna menekan tombol <i>NO</i> , pengguna akan kembali memainkan permainan
Hasil uji coba	Berhasil
Kondisi akhir	<ul style="list-style-type: none"> • Jika pengguna menekan <i>YES</i>, pengguna akan berada di <i>main menu</i>. • Jika pengguna menekan <i>NO</i> pengguna akan berada di layar permainan “Get The Ball”.



Gambar 5.1 Tampilan Saat Uji Coba Memainkan Permainan "Get The Ball"

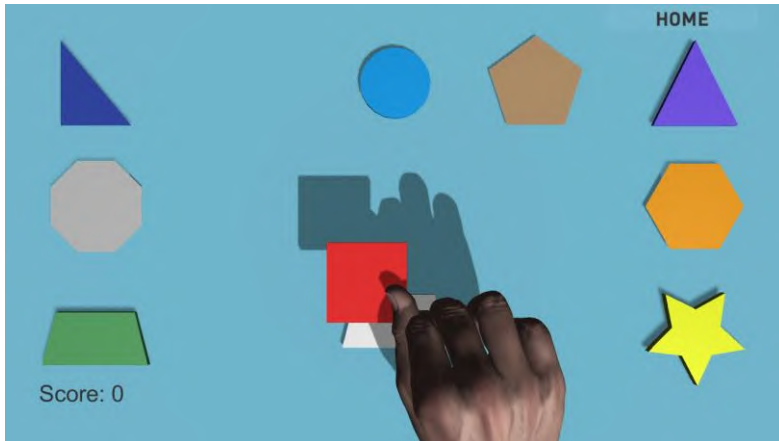
5.2.3.2. Uji Coba Memainkan Permainan “Match The Shape”

Pengujian dalam memainkan permainan “Match The Shape” ini dilakukan untuk menguji keberhasilan fungsionalitas utama dari permainan ini. Hasil dari pengujian ini akan secara lengkap dijabarkan pada Tabel 5.3. Hasil dari skenario uji coba tersebut dapat dilihat pada Gambar 5.2.

Tabel 5.3 Hasil Uji Coba Memainkan Permainan “Match The Shape”

ID	UJ- 002
Nama	Uji Coba Memainkan Permainan “Match The Shape”
Tujuan uji coba	Menguji fitur untuk memainkan permainan “Match The Shape”
Kondisi awal	Proses <i>rendering</i> objek sudah selesai
<i>Skenario 1</i>	<i>Pengguna mengambil bangun datar dan meletakan nya diatas bangun datar display</i>
Masukan	Meletakkan bangun datar di atas bangun datar <i>display</i>

ID	UJ- 002
Keluaran yang diharapkan	Skor bertambah dan bangun datar <i>display</i> yang baru muncul
Hasil uji coba	Berhasil
Kondisi akhir	<ul style="list-style-type: none"> • Jika permainan sudah mencapai skor maksimal, pengguna akan dimunculkan layar <i>Game Over</i>. • Jika permainan belum mencapai skor maksimal, pengguna akan kembali berada di layar permainan “Match The Shape”.
<i>Skenario 2</i>	<i>Pengguna meletakkan bangun datar yang berbeda ke bangun datar display</i>
Masukan	Meletakkan bangun datar di atas bangun datar <i>display</i>
Keluaran yang diharapkan	Skor tetap dan bangun datar <i>display</i> yang baru muncul
Hasil uji coba	Berhasil
Kondisi akhir	Pengguna akan berada di layar permainan “Match The Shape”
<i>Skenario 3</i>	<i>Pengguna menekan tombol HOME pada Halaman permainan</i>
Masukan	Klik tombol HOME
Keluaran yang diharapkan	Pengguna akan dimunculkan menu untuk konfirmasi apakah benar akan kembali ke <i>homescreen</i> atau tidak. Jika pengguna menekan <i>YES</i> , pengguna akan kembali ke <i>main menu</i> , jika pengguna menekan tombol <i>NO</i> , pengguna akan kembali memainkan permainan
Hasil uji coba	Berhasil
Kondisi akhir	<ul style="list-style-type: none"> • Jika pengguna menekan <i>YES</i>, pengguna akan berada di <i>main menu</i>. • Jika pengguna menekan <i>NO</i>, pengguna akan berada di layar permainan “Match The Shape”.



Gambar 5.2 Tampilan Saat Uji Coba Memainkan Permainan "Match The Shape"

Pada pengujian ini untuk fungsionalitas utama dalam permainan seperti meletakkan bangun datar ke atas bangun datar *display* dan menambahkan skor jika benar sudah dapat berfungsi dengan baik. Namun dalam penggunaan LMC pada permainan ini terkadang masih ditemukan kendala pada saat memainkan permainan seperti susah mengambil objek yang ada dipinggir layar permainan. Hal ini dikarenakan area permainan yang ditampilkan terlalu luas sehingga LMC yang sensornya diletakkan di tengah area permainan, sulit dalam menjangkau objek yang ada dipinggir layar tersebut. Untuk mengatasinya mungkin dapat dikurangi jumlah dari bangun datar sehingga area permainan tidak terlalu luas. Dalam mengambil objek juga kadang masih ditemukan kesulitan, hal ini dikarenakan LMC lebih diperuntukkan untuk digunakan pada permainan yang tiga dimensi sehingga sulit ketika digunakan untuk permainan dua dimensi seperti pada permainan ini.

5.2.3.3. Uji Coba Memainkan Permainan "Find The Matching Animals"

Pengujian dalam memainkan permainan "Find The Matching Animals" ini dilakukan untuk menguji keberhasilan fungsionalitas

utama dari permainan ini. Hasil dari pengujian ini akan secara lengkap dijabarkan pada Tabel 5.4. Hasil dari skenario uji coba tersebut dapat dilihat pada Gambar 5.3.

Tabel 5.4 Hasil Uji Coba Memainkan Permainan “Find The Matching Animals”

ID	UJ- 003
Nama	Uji Coba Memainkan Permainan “Find The Matching Animals”
Tujuan uji coba	Menguji fitur untuk memainkan permainan “Find The Matching Animals”
Kondisi awal	Proses <i>rendering</i> objek sudah selesai
<i>Skenario 1</i>	<i>Pengguna mengambil dua binatang yang sama dan dimasukkan ke dua kandang yang tersedia</i>
Masukan	Meletakkan binatang ke dua kandang yang tersedia
Keluaran yang diharapkan	Skor bertambah dan empat binatang yang baru muncul
Hasil uji coba	Berhasil
Kondisi akhir	<ul style="list-style-type: none"> • Jika permainan sudah mencapai skor maksimal, pengguna akan dimunculkan layar <i>Game Over</i>. • Jika permainan belum mencapai skor maksimal, pengguna akan kembali berada di layar permainan “Find The Matching Animals”.
<i>Skenario 2</i>	<i>Pengguna mengambil dua binatang yang berbeda dan dimasukkan ke dua kandang yang tersedia</i>
Masukan	Meletakkan binatang ke dua kandang yang tersedia
Keluaran yang diharapkan	Skor tetap dan empat binatang yang baru muncul
Hasil uji coba	Berhasil
Kondisi akhir	Pengguna akan berada di layar permainan “Find The Matching Animals”.
<i>Skenario 3</i>	<i>Pengguna menekan tombol HOME pada Halaman permainan</i>

ID	UJ- 003
Masukan	Klik tombol HOME
Keluaran yang diharapkan	Pengguna akan dimunculkan menu untuk konfirmasi apakah benar akan kembali ke <i>homescreen</i> atau tidak. Jika pengguna menekan <i>YES</i> , pengguna akan kembali ke <i>main menu</i> , jika pengguna menekan tombol <i>NO</i> , pengguna akan kembali memainkan permainan
Hasil uji coba	Berhasil
Kondisi akhir	<ul style="list-style-type: none"> • Jika pengguna menekan <i>YES</i>, pengguna akan berada di <i>main menu</i>. • Jika pengguna menekan <i>NO</i>, pengguna akan berada di layar permainan “Find The Matching Animals”.

Pada pengujian ini untuk fungsionalitas utama dalam permainan seperti meletakkan binatang ke dalam kedua kandang yang disediakan dan menambahkan skor jika benar sudah dapat berfungsi dengan baik. Namun dalam penggunaan LMC pada permainan ini terkadang masih ditemukan kendala pada saat memainkan permainan seperti susah mengambil objek yang ada dipinggir layar permainan. Hal ini dikarenakan area permainan yang ditampilkan terlalu luas sehingga LMC yang sensornya diletakkan di tengah area permainan, sulit dalam menjangkau objek yang ada dipinggir layar tersebut. Setelah dilakukan percobaan dengan menggunakan perangkat keras yang berbeda, performa LMC dipengaruhi juga oleh perangkat keras yang menjalankannya. Ketika menjalankan dengan perangkat keras yang lebih baik, kesulitan ketika menjangkau objek menjadi tidak seburuk ketika menggunakan perangkat keras yang biasa saja.



Gambar 5.3 Tampilan Saat Uji Coba Memainkan Permainan "Find The Matching Animals"

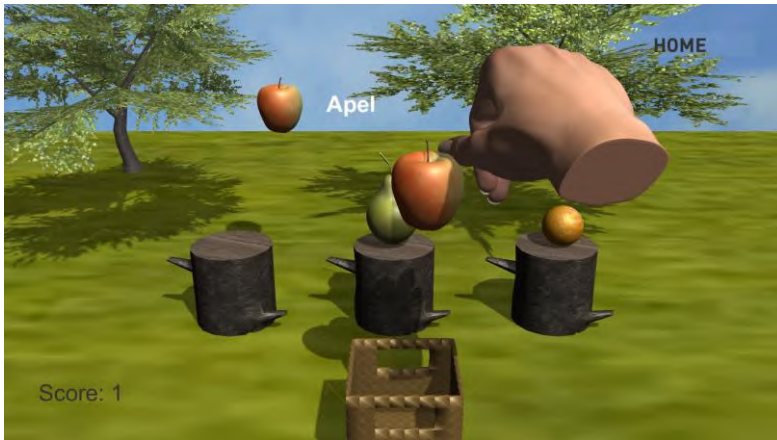
5.2.3.4. Uji Coba Memainkan Permainan "Pick The Fruit"

Pengujian dalam memainkan permainan "Pick The Fruit" ini dilakukan untuk menguji keberhasilan fungsionalitas utama dari permainan ini. Hasil dari pengujian ini akan secara lengkap dijabarkan pada Tabel 5.5. Hasil dari skenario uji coba tersebut dapat dilihat pada Gambar 5.4.

Tabel 5.5 Hasil Uji Coba Memainkan Permainan "Pick The Fruit"

ID	UJ- 004
Nama	Uji Coba Memainkan Permainan "Pick The Fruit"
Tujuan uji coba	Menguji fitur untuk memainkan permainan "Pick The Fruit"
Kondisi awal	Proses <i>rendering</i> objek sudah selesai
<i>Skenario 1</i>	<i>Pengguna mengambil buah yang sama dengan display yang ditampilkan dan dimasukkan ke keranjang yang tersedia</i>
Masukan	Meletakkan binatang ke dua kandang yang tersedia

ID	UJ- 004
Keluaran yang diharapkan	Skor bertambah dan empat binatang yang baru muncul
Hasil uji coba	Berhasil
Kondisi akhir	<ul style="list-style-type: none"> Jika permainan sudah mencapai skor maksimal, pengguna akan dimunculkan layar <i>Game Over</i>. Jika permainan belum mencapai skor maksimal, pengguna akan kembali berada di layar permainan “Pick The Fruit”.
Skenario 2	<i>Pengguna mengambil dua binatang yang berbeda dan dimasukkan ke dua kandang yang tersedia</i>
Masukan	Meletakkan binatang ke dua kandang yang tersedia
Keluaran yang diharapkan	Skor tetap dan empat binatang yang baru muncul
Hasil uji coba	Berhasil
Kondisi akhir	Pengguna akan berada di layar permainan “Pick The Fruit”.
Skenario 3	<i>Pengguna menekan tombol HOME pada Halaman permainan</i>
Masukan	Klik tombol HOME
Keluaran yang diharapkan	Pengguna akan dimunculkan menu untuk konfirmasi apakah benar akan kembali ke <i>homescreen</i> atau tidak. Jika pengguna menekan <i>YES</i> , pengguna akan kembali ke <i>main menu</i> , jika pengguna menekan tombol <i>NO</i> , pengguna akan kembali memainkan permainan
Hasil uji coba	Berhasil
Kondisi akhir	<ul style="list-style-type: none"> Jika pengguna menekan <i>YES</i>, pengguna akan berada di <i>main menu</i>. Jika pengguna menekan <i>NO</i>, pengguna akan berada di layar permainan “Pick The Fruit”.



Gambar 5.4 Tampilan Saat Uji Coba Memainkan Permainan "Pick The Fruit"

5.2.4. Pengujian oleh anak penderita autisme

Pada subbab ini dijelaskan uji coba perangkat lunak yang dilakukan oleh anak-anak penderita autisme. Peserta dari pengujian aplikasi ini berumur 4-12 tahun, anak-anak ini adalah anak-anak berkebutuhan khusus yang didik oleh Yayasan Alejo Pelita Indah Surabaya. Metode pengujian yang akan dilakukan terbagi menjadi dua metode yaitu skenario kuesioner dan pendataan performa permainan.

5.2.4.1. Skenario Kuesioner

Pada subbab ini dijelaskan metode pertama yaitu skenario uji coba dengan kuesioner untuk pengujian performa dari permainan-permainan serta kontrol antarmuka permainan dengan menggunakan Leap Motion Controller dari aplikasi ini, pengujian dan tingkat kenyamanan aplikasi dengan melibatkan anak-anak yang memiliki kebutuhan khusus pada Yayasan Alejo Pelita Indah. Metode kuesioner yang digunakan adalah *Likert-type scale* dengan skala 1-6.

Skenario yang diuji pengguna adalah memainkan keempat permainan yang disediakan dan kemudian pengguna akan

ditanyakan pertanyaan mengenai permainan berdasarkan kuesioner yang dibuat. Kuesioner pengguna dapat dilihat pada Tabel 5.6.

Tabel 5.6 Kuesioner Pengguna

NO	PARAMETER	SKALA NILAI					
		STS	TS	KS	CS	S	SS
		1	2	3	4	5	6
TAMPILAN							
1.	Aplikasi memiliki tampilan, desain dan warna yang menarik						
2.	Aplikasi memiliki tampilan yang mudah dipahami						
3.	Aplikasi memiliki tata letak tombol yang mudah dikenali						
KENYAMANAN							
1.	Aplikasi mudah dipahami saat digunakan						
2.	Petunjuk permainan yang diberikan dapat membantu memahami permainan						
3.	Pengguna bingung saat memainkan aplikasi						
PERMAINAN							
1.	Aplikasi menyenangkan untuk dimainkan						
2.	Permainan pada aplikasi mudah dipahami						

5.2.4.2. Pendataan Performa Pemain

Pada subbab ini dijelaskan metode kedua yaitu pendataan performa pemain dengan parameter skor dan waktu penyelesaian bermain, hal ini dilakukan untuk melakukan pengujian keefektifan aplikasi dalam membantu meningkatkan kemampuan motorik halus dari anak-anak yang autisme atau memiliki kebutuhan khusus.

Skenario yang dilakukan ketika melakukan pendataan ini adalah pertama pemain akan dibiarkan bermain dengan arahan di awal permainan dari petunjuk yang ada pada petunjuk permainan dan diberikan contoh memainkan permainan yang dicontohkan oleh penulis. Setelah selesai bermain, pemain akan bermain kembali namun sekarang dengan bimbingan tidak hanya saat awal permainan namun ketika bermain pun anak akan terus dibimbing. Dari skenario ini akan dapat dilihat keefektifan permainan serta jika

permainan dilakukan secara rutin akan dapat dilihat pula perubahan performa motorik halus dari tiap anak.

5.2.5. Hasil Uji Coba

Pada subbab ini dijabarkan hasil uji coba perangkat lunak yang dilakukan oleh anak-anak yang memiliki kebutuhan khusus. peserta dari pengujian aplikasi ini berumur 4-12 tahun, anak-anak ini adalah anak-anak berkebutuhan khusus yang didik oleh Yayasan Alejo Pelita Indah Surabaya. Hasil dari uji coba ini terbagi menjadi tiga bagian, pertama adalah hasil pengisian kuesioner, hasil pendataan performa permainan dan juga tanggapan dari pengguna.

5.2.5.1. Hasil Kuesioner

Pengisian kuesioner melibatkan empat orang guru dari Yayasan Alejo Pelita Indah Surabaya yang memandu anak-anak yang memiliki kebutuhan khusus dalam memainkan aplikasi ini, guru-guru ini mewakili anak-anak dalam menjawab kuesioner, pendapat guru-guru ini didapatkan dari pengamatan mereka ketika melihat anak-anak didik mereka yang memainkan aplikasi ini. Setiap poin penilaian memiliki skala nilai 1 sampai dengan 6 dengan keterangan 1 adalah sangat tidak setuju, 2 adalah tidak setuju, 3 adalah kurang setuju, 4 adalah cukup setuju, 5 adalah setuju, dan 6 adalah sangat setuju. Hasil dari kuesioner dapat dilihat pada Tabel 5.7.

Tabel 5.7 Hasil Kuesioner

NO	PARAMETER	NILAI RATA RATA
TAMPILAN		
1.	Aplikasi memiliki tampilan, desain dan warna yang menarik	4.25
2.	Aplikasi memiliki tampilan yang mudah dipahami	4.25
3.	Aplikasi memiliki tata letak tombol yang mudah dikenali	4
KENYAMANAN		
1.	Aplikasi mudah dipahami saat digunakan	4.25
2.	Petunjuk permainan yang diberikan dapat membantu memahami permainan	4.5

NO	PARAMETER	NILAI RATA RATA
3.	Pengguna bingung saat memainkan aplikasi	4.25
PERMAINAN		
1.	Aplikasi menyenangkan untuk dimainkan	5
2.	Permainan pada aplikasi mudah dipahami	4.25

Dari hasil kuesioner pada Tabel 5.7 didapatkan nilai rata-rata penilaian terhadap tampilan, kenyamanan dan permainan. Dari penilaian pada tampilan, pengguna cukup setuju aplikasi memiliki tampilan dan desain yang menarik, pengguna juga cukup setuju bahwa aplikasi memiliki tampilan yang mudah dipahami dan pengguna cukup setuju pada pendapat tata letak tombol dari aplikasi mudah dikenali.

Dari penilaian pada kenyamanan, pengguna cukup setuju bahwa aplikasi dapat mudah dipahami ketika digunakan, pengguna cukup setuju jika petunjuk-petunjuk yang diberikan pada awal permainan dapat membantu dalam memahami permainan. Pengguna juga cukup setuju bahwa pengguna bingung saat memainkan aplikasi.

Dari penilaian pada permainan, pengguna setuju bahwa aplikasi menyenangkan untuk dimainkan. Selain itu, pengguna cukup setuju bahwa permainan-permainan pada aplikasi ini mudah dipahami.

5.2.5.2. Hasil Pendataan Performa Permainan

Pendataan ini dilakukan dengan mengambil performa dari anak-anak yang memainkan permainan-permainan pada aplikasi ini. Pendataan yang dilakukan memiliki dua Skenario, skenario pertama (S1) anak akan diberikan instruksi pada awal permainan dari petunjuk permainan yang diberikan dan dengan dicontohkan cara bermainnya namun setelah dicontohkan anak akan bermain sendiri, selesai bermain hasil skor dan waktu penyelesaian dari tiap permainan akan dicatat. Skenario kedua (S2) adalah anak akan

dibimbing terus ketika bermain. Data performa bermain dapat dilihat pada Tabel 5.8 - Tabel 5.11.

Tabel 5.8 Data Performa Pemain Permainan “Get The Ball”

NO	PEMAIN	PERMAINAN “GET THE BALL”			
		S1		S2	
		SKOR	WAKTU	SKOR	WAKTU
1.	ANAK 1	1	3 Menit	5	49 Detik
2.	ANAK 2	0	3 Menit	3	3 Menit
3.	ANAK 3	4	3 Menit	5	50 Detik
4.	ANAK 4	5	6 Menit	5	2 Menit

Dari Tabel 5.8 yang merupakan hasil pendataan yang dilakukan saat memainkan permainan “Get The Ball”, dapat terlihat beberapa anak masih sulit dalam memainkan permainan tanpa dibimbing. Ketika bermain dengan bimbingan, skor dan waktu penyelesaian dari tiap anak naik dengan cukup signifikan.

Tabel 5.9 Data Performa Pemain Permainan “Match The Shape”

NO	PEMAIN	PERMAINAN “MATCH THE SHAPE”			
		S1		S2	
		SKOR	WAKTU	SKOR	WAKTU
1.	ANAK 1	2	3 menit	3	3 menit
2.	ANAK 2	0	3 menit	1	3 menit
3.	ANAK 3	3	3 menit	5	2 menit
4.	ANAK 4	3	6 menit	5	5 menit

Dari Tabel 5.9 yang merupakan hasil pendataan yang dilakukan saat memainkan permainan “Match The Shape”, dapat terlihat beberapa anak masih sulit dalam memainkan permainan tanpa dibimbing maupun dengan bimbingan. Mayoritas dari anak-anak yang memainkan permainan ini tidak dapat menyelesaikan tujuan dari permainan dari waktu yang diberikan.

Tabel 5.10 Data Performa Pemain Permainan “Find The Matching Animals”

NO	PEMAIN	PERMAINAN “FIND THE MATCHING ANIMALS”			
		S1		S2	
		SKOR	WAKTU	SKOR	WAKTU
1.	ANAK 1	0	3 menit	1	3 menit
2.	ANAK 2	0	3 menit	1	3 menit
3.	ANAK 3	0	3 menit	2	3 menit

NO	PEMAIN	PERMAINAN "FIND THE MATCHING ANIMALS"			
		S1		S2	
		SKOR	WAKTU	SKOR	WAKTU
4.	ANAK 4	0	6 menit	2	6 menit

Dari Tabel 5.10 yang merupakan hasil pendataan yang dilakukan saat memainkan permainan "Find The Matching Animals", dapat terlihat beberapa anak masih sulit dalam memainkan permainan tanpa dibimbing maupun dengan bimbingan. Mayoritas dari anak-anak yang memainkan permainan ini tidak dapat menyelesaikan tujuan dari permainan dari waktu yang diberikan.

Tabel 5.11 Data Performa Pemain Permainan "Pick The Fruit"

NO	PEMAIN	PERMAINAN "PICK THE FRUIT"			
		S1		S2	
		SKOR	WAKTU	SKOR	WAKTU
1.	ANAK 1	3	3 menit	5	47 detik
2.	ANAK 2	0	3 menit	3	3 menit
3.	ANAK 3	3	3 menit	5	50 detik
4.	ANAK 4	5	65 detik	5	91 detik

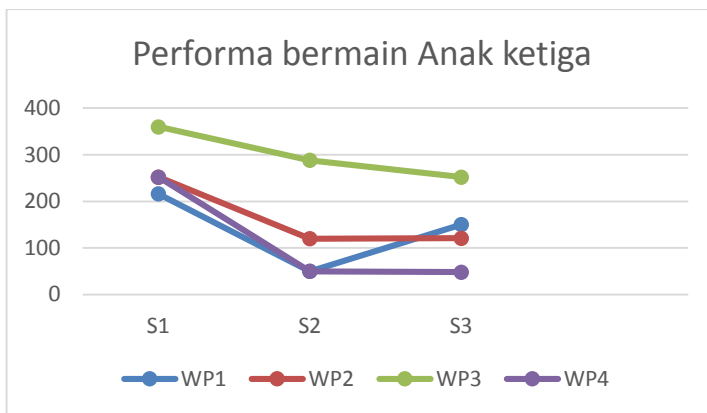
Dari Tabel 5.11 yang merupakan hasil pendataan yang dilakukan saat memainkan permainan "Pick The Fruit", dapat terlihat mayoritas dari anak-anak yang memainkan dapat memainkan permainan dengan baik tanpa bimbingan maupun dengan bimbingan. Ketika bermain dengan bimbingan anak dapat menyelesaikan tujuan dari permainan dengan lebih cepat dibandingkan tanpa bimbingan ketika bermain.

Pada uji coba ini penulis hanya mencoba permainan ini pada tiap anak satu kali pada setiap skenario, namun penulis berkesempatan untuk mencoba kembali permainan ini kepada anak ketiga. Hal ini dilakukan untuk melihat perkembangan dari performa pemain ketika memainkan permainan ini untuk yang kedua kalinya. Dalam memainkan permainan ini untuk yang kedua kalinya, penulis hanya memberikan instruksi saja seperti pada skenario pertama tanpa membimbing anak lebih lanjut. Hasil dari

percobaan ini dapat dilihat pada tabel yang ada pada Tabel 5.12. Percobaan ketiga ini akan dinamakan skenario 3, perbandingan ketiga skenario yang dilakukan akan dibandingkan pada grafik yang ada pada Gambar 5.5.

Tabel 5.12 Data Performa Anak ke-3 pada Skenario 3

NO	PERMAINAN	SKENARIO 3	
		SKOR	WAKTU
1.	"GET THE BALL"	5	150 detik
2.	"MATCH THE SHAPE"	5	121 detik
3.	"FIND THE MATCHING ANIMALS"	3	3 menit
4.	"PICK THE FRUIT"	5	48 detik



Gambar 5.5 Grafik performa bermain anak ketiga

Pada grafik di atas S1, S2 dan S3 merepresentasikan Skenario 1-3 yang di uji coba, sedangkan WP 1, WP2, WP3, dan WP4 merepresentasikan waktu bermain dari anak dari tiap permainan dalam bentuk detik. Karena pada beberapa permainan anak ketiga tidak dapat menyelesaikan tujuan dari permainan hingga mencapai skor 5, maka pada waktu permainan akan ditambahkan dengan penalti, penalti ini adalah merupakan hasil dari pengurangan skor yang perlu dicapai yaitu 5 dan skor yang didapatkan lalu dikalikan dengan 36 detik, 36 detik ini adalah hasil pembagian waktu

permainan yaitu 180 detik atau 3 menit dengan 5. Nilai contohnya pada WP1, waktu permainan pada skenario 1 nya menjadi 216 detik, karena pada permainan “Get The Ball” pemain berhasil mendapatkan skor 4 dalam 3 menit, karena pemain masih kurang 1 skor dari skor maksimal maka WP1 nya dari 3 menit atau 180 detik ditambahkan 36 detik menjadi 216 detik.

Dari grafik pada Gambar 5.5 dapat dilihat perkembangan pemain ketika mencoba memainkan permainan-permainan ini tanpa bantuan. Pemain pada permainan “Get The Ball” pertama kali dapat mendapatkan skor 4 dalam 3 menit, setelah memainkannya kembali pemain dapat mendapatkan skor 5 dalam 150 detik, lalu pada permainan “Match The Shape” pertama kali dapat mendapatkan skor 3 dalam 3 menit, setelah memainkannya kembali pemain dapat mendapatkan skor 5 dalam 121 detik, sedangkan pada permainan “Find The Matching Animals” pertama kali tidak dapat mendapatkan skor dalam 3 menit, setelah memainkannya kembali pemain dapat mendapatkan skor 3 dalam 3 menit, pada permainan terakhir “Pick The Fruit” pertama kali dapat mendapatkan skor 3 dalam 3 menit, setelah memainkannya kembali pemain dapat mendapatkan skor 5 dalam 48 detik. Dari hasil ini dapat dilihat perkembangan dari anak ketiga dalam memainkan permainan-permainan dalam aplikasi ini. Pada permainan “Pick The Fruit” anak ketiga bahkan bisa menyelesaikan permainan dengan lebih baik dari ketika anak ketiga bermain dengan bimbingan pada skenario 2.

5.2.5.3. Tanggapan Pengguna

Pada kuesioner terdapat tanggapan pengguna berupa kritik dan saran untuk permainan-permainan pada aplikasi ini. Pada subbab ini ditunjukkan tanggapan responden setelah menggunakan aplikasi. Detail tanggapan akan ditampilkan pada Tabel 5.13.

Tabel 5.13 Tanggapan Pengguna

No.	Tanggapan
1.	<i>Game</i> Hewan bentuknya terlalu kecil sehingga anak kesusahan untuk menggenggam / menjimpit. Untuk <i>Game</i> yang lain cukup baik dalam melatih motorik halus pada anak.
2.	Mungkin bisa dengan menunjuk (bukan genggam / jimpit) untuk beberapa Permainan.
3.	Metode ini cukup membantu anak untuk melatih ketahanan konsentrasi dan kefokuskan anak. Dan bisa membantu untuk melatih gerak motorik halus anak. Namun masih banyak yang harus ditambahi pada aplikasi <i>game</i> ini dengan diberi <i>reward</i> agar lebih menarik.
4.	Tidak ada tanggapan khusus.

5.3. Evaluasi Pengujian

Subbab ini membahas mengenai evaluasi terhadap pengujian-pengujian yang telah dilakukan. Dalam hal ini, evaluasi menunjukkan data rekapitulasi dari hasil pengujian yang telah dilakukan sebelumnya. Berdasarkan hasil pengujian fungsional oleh penulis, pengujian kegunaan oleh pengguna berdasarkan kuesioner dan pendataan performa yang telah dilaksanakan dijelaskan pada subbab sebelumnya, maka didapatkan evaluasi sebagai berikut:

1. Aplikasi berhasil menjalankan fungsionalitas utama dari keempat permainan dengan baik dapat dilihat pada UJ-001 sampai UJ-004.
2. Berdasarkan kuesioner, penilaian untuk tampilan aplikasi sudah cukup menarik dan mudah dipahami namun dapat ditingkatkan kembali. Hal ini dibuktikan dengan hasil kuesioner yang dapat dilihat pada Tabel 5.7.

3. Pengguna menilai Tingkat kenyamanan saat memainkan permainan-permainan ini masih perlu ditingkatkan lagi. Hal ini dibuktikan dengan hasil kuesioner yang dapat dilihat pada Tabel 5.7.
4. Permainan-permainan dari aplikasi ini cukup menyenangkan dan mudah dipahami. Hal ini dibuktikan dengan hasil kuesioner yang dapat dilihat pada Tabel 5.7.
5. Permainan “Get The Ball” dan “Pick The Fruit” sudah dapat dimainkan dengan cukup baik oleh anak-anak yang memiliki kebutuhan khusus, namun pada permainan “Find The Matching Animals” dan Match The Shape” anak-anak masih merasa kesulitan dalam menyelesaikan tujuan dari permainan dari waktu yang diberikan.
6. Bentuk objek dari permainan terlalu kecil “Find The Matching Animals”, kontrol yang perlu lebih dimudahkan dan juga perlu adanya *reward* setelah memainkan permainan merupakan beberapa tanggapan yang didapat dalam menjalankan permainan ini. Hal ini dibuktikan dengan tanggapan pengguna yang dapat dilihat pada Tabel 5.13

[Halaman ini sengaja dikosongkan]

LAMPIRAN

TUGAS AKHIR – 5112100123 AHMAD FATHORIQ
RANCANG BANGUN APLIKASI UNTUK MEMPERBAIKI KEMAMPUAN MOTORIK HALUS PADA ANAK
PENDERITA AUTISME DENGAN MENGGUNAKAN TEKNOLOGI LEAP MOTION CONTROLLER
KUISIONER SURVEY PENGGUNA

IDENTITAS KORESPONDEN

Nama : Hartono Jenis Kelamin L/P
Usia : 5 tahun

A. KARAKTERISTIK KORESPONDEN

1. Pernahkah anda menggunakan leap motion controller?
 - a. Lebih dari 10 kali
 - b. 6-10 kali
 - c. 2-5 kali
 - d. 1 kali
2. Apakah Anda pernah melakukan terapi okupasi (jika Tidak Pernah, lewatkan pertanyaan no.3)?
 - a. Pernah
 - b. Tidak Pernah
3. Seberapa sering Anda melakukan terapi?
 - a. Lebih dari 5 kali
 - b. 2 – 5 kali
 - c. 1 kali
 - d. Tidak pernah

B. PENILAIAN TERHADAP APLIKASI


Isilah tabel dibawah ini dengan menggunakan tanda (v)

STS = Sangat Tidak Setuju TS = Tidak Setuju KS = Kurang Setuju CS = Cukup Setuju S = Setuju SS = Sangat Setuju

		SKALA NILAI					
NO	PARAMETER	STS	TS	KS	CS	S	SS
		1	2	3	4	5	
TAMPILAN							
1.	Aplikasi memiliki tampilan, desain dan warna yang menarik					✓	
2.	Aplikasi memiliki tampilan yang mudah dipahami					✓	
3.	Aplikasi memiliki tata letak tombol yang mudah dikenali					✓	
KENYAMANAN							
1.	Aplikasi mudah dipahami saat digunakan						✓
2.	Pentunjuk permainan yang diberikan dapat membantu memahami permainan						✓
3.	Pengguna bingung saat memainkan aplikasi					✓	
PERMAINAN							
1.	Aplikasi menyenangkan untuk dimainkan					✓	✓
2.	Permainan pada aplikasi mudah dipahami					✓	

C. KRITIK DAN SARAN

Surabaya, 10 - 06 - 2016


 MS. YANA

Gambar A.1 Kuesioner Responden Pertama

TUGAS AKHIR – 5112100123 AHMAD FATHORIQ
RANCANG BANGUN APLIKASI UNTUK MEMPERBAIKI KEMAMPUAN MOTORIK HALUS PADA ANAK
PENDERITA AUTISME DENGAN MENGGUNAKAN TEKNOLOGI LEAP MOTION CONTROLLER
KUISIONER SURVEY PENGGUNA

IDENTITAS KORESPONDEN

Nama : DATTEN
 Usia : 4 tahun

Jenis Kelamin LP

A. KARAKTERISTIK KORESPONDEN

1. Pernahkah anda menggunakan leap motion controller?
 - a. Lebih dari 10 kali
 - b. 6-10 kali
 - c. 2-5 kali
 - d. 1 kali
 - e) Tidak Pernah
2. Apakah Anda pernah melakukan terapi okupasi (Jika Tidak Pernah, lewatkan pertanyaan no.3)?
 - a. Pernah
 - b) Tidak Pernah
3. Seberapa sering Anda melakukan terapi?
 - a) Lebih dari 5 kali
 - b. 2 – 5 kali
 - c. 1 kali
 - d. Tidak pernah

B. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (v)

STS = Sangat Tidak Setuju TS = Tidak Setuju KS = Kurang Setuju CS = Cukup Setuju S = Setuju SS = Sangat Setuju

NO	PARAMETER	SKALA NILAI					
		STS	TS	KS	CS	S	SS
		1	2	3	4	5	6
TAMPILAN							
1.	Aplikasi memiliki tampilan, desain dan warna yang menarik				✓		
2.	Aplikasi memiliki tampilan yang mudah dipahami				✓		
3.	Aplikasi memiliki tata letak tombol yang mudah dikenali			✓			
KENYAMANAN							
1.	Aplikasi mudah dipahami saat digunakan				✓		
2.	Pentunjuk permainan yang diberikan dapat membantu memahami permainan			✓			
3.	Pengguna bingung saat memainkan aplikasi					✓	
PERMAINAN							
1.	Aplikasi menyenangkan untuk dimainkan				✓		
2.	Permainan pada aplikasi mudah dipahami				✓		

C. KRITIK DAN SARAN

Metode ini cukup membantu u/ melatih ketahanan konsentrasi & kefokusn anak. Dan bisa membantu u/ melatih gerak motorik halus anak. Namun masih banyak yg harus ditambahi pd aplikasi game ini dg diberi reward agar lebih menarik.

Surabaya, 09 - Juni - 2016

(Signature)
 Dina

Gambar A.2 Kuesioner Responden Kedua

TUGAS AKHIR – 5112100123 AHMAD FATHORIQ
RANCANG BANGUN APLIKASI UNTUK MEMPERBAIKI KEMAMPUAN MOTORIK HALUS PADA ANAK
PENDERITA AUTISME DENGAN MENGGUNAKAN TEKNOLOGI LEAP MOTION CONTROLLER
KUISIONER SURVEY PENGGUNA

IDENTITAS KORESPONDEN

Nama : Yusuf Alif
 Usia : 6 tahun

Jenis Kelamin : L/P

A. KARAKTERISTIK KORESPONDEN

1. Pernahkah anda menggunakan leap motion controller?
 - a. Lebih dari 10 kali
 - b. 6-10 kali
 - c. 2-5 kali
 - d. 1 kali
 - e. Tidak Pernah
2. Apakah Anda pernah melakukan terapi okupasi (Jika Tidak Pernah, lewatkan pertanyaan no.3)?
 - a. Pernah
 - b. Tidak Pernah
3. Seberapa sering Anda melakukan terapi?
 - a. Lebih dari 5 kali
 - b. 2 – 5 kali
 - c. 1 kali
 - d. Tidak pernah

B. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (v)

STS = Sangat Tidak Setuju TS = Tidak Setuju KS = Kurang Setuju CS = Cukup Setuju S = Setuju SS = Sangat Setuju

NO	PARAMETER	SKALA NILAI					
		STS	TS	KS	CS	S	SS
		1	2	3	4	5	6
TAMPILAN							
1.	Aplikasi memiliki tampilan, desain dan warna yang menarik				✓		
2.	Aplikasi memiliki tampilan yang mudah dipahami				✓		
3.	Aplikasi memiliki tata letak tombol yang mudah dikenali				✓		
KENYAMANAN							
1.	Aplikasi mudah dipahami saat digunakan			✓			
2.	Pentunjuk permainan yang diberikan dapat membantu memahami permainan				✓		
3.	Pengguna bingung saat memainkan aplikasi			✓			
PERMAINAN							
1.	Aplikasi menyenangkan untuk dimainkan				✓		
2.	Permainan pada aplikasi mudah dipahami			✓			

C. KRITIK DAN SARAN

Mengen bisa dg lanjut (bukan hanya gambar / gambar)
 untuk beberapa permainan.

Surabaya, 09.6.2016

(Alif)
 anak.

Gambar A.3 Kuesioner Responden Ketiga

TUGAS AKHIR – 5112100123 AHMAD FATHORIQ
RANCANG BANGUN APLIKASI UNTUK MEMPERBAIKI KEMAMPUAN MOTORIK HALUS PADA ANAK
PENDERITA AUTISME DENGAN MENGGUNAKAN TEKNOLOGI LEAP MOTION CONTROLLER
KUISIONER SURVEY PENGGUNA

IDENTITAS KORESPONDEN

Nama : Clement

Jenis Kelamin (L) P

Usia : 12 tahun

A. KARAKTERISTIK KORESPONDEN

1. Pernahkah anda menggunakan leap motion controller?
 - a. Lebih dari 10 kali
 - b. 6-10 kali
 - c. 2-5 kali
 - d. 1 kali
 - (a) Tidak Pernah
2. Apakah Anda pernah melakukan terapi okupasi (Jika Tidak Pernah, lewatkan pertanyaan no.3)?
 - (a) Pernah
 - b. Tidak Pernah
3. Seberapa sering Anda melakukan terapi?
 - (a) Lebih dari 5 kali
 - b. 2 – 5 kali
 - c. 1 kali
 - d. Tidak pernah

B. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (v)

STS = Sangat Tidak Setuju TS = Tidak Setuju KS = Kurang Setuju CS = Cukup Setuju S = Setuju SS = Sangat Setuju

NO	PARAMETER	SKALA NILAI					
		STS	TS	KS	CS	S	SS
		1	2	3	4	5	6
TAMPILAN							
1.	Aplikasi memiliki tampilan, desain dan warna yang menarik						
2.	Aplikasi memiliki tampilan yang mudah dipahami					✓	
3.	Aplikasi memiliki tata letak tombol yang mudah dikenali					✓	
KENYAMANAN							
1.	Aplikasi mudah dipahami saat digunakan				✓		
2.	Pentunjuk permainan yang diberikan dapat membantu memahami permainan					✓	
3.	Pengguna bingung saat memainkan aplikasi			✓			
PERMAINAN							
1.	Aplikasi menyenangkan untuk dimainkan				✓		
2.	Permainan pada aplikasi mudah dipahami				✓		

C. KRITIK DAN SARAN

Game hewan bentuknya terlalu kecil sehingga anak
 kesulitan untuk menggenggam / menjepit.
 Untuk game yang lain cukup baik dan melatih motorik halus
 pada anak.

Surabaya, 09 Juni 2016

( Miss Hong)

Gambar A.4 Kuesioner Responden Keempat

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan Tugas Akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1. Kesimpulan

Dari hasil selama proses perancangan, implementasi, serta pengujian dapat diambil kesimpulan sebagai berikut:

1. Dari tanggapan yang diberikan pada kuesioner uji coba, pilihan permainan dari aplikasi ini cukup membantu anak dalam melatih ketahanan konsentrasi dan melatih kemampuan motorik halus dari anak-anak berkebutuhan khusus.
2. Gestur menggenggam atau menjepit dengan ujung telunjuk dan ibu jari yang dimiliki oleh Leap Motion Controller dapat meningkatkan kemampuan motorik halus dari anak.
3. Perangkat keras mempengaruhi performa Leap Motion Controller dalam menjalankan permainan. Dengan menggunakan perangkat keras yang lebih baik, permainan akan lebih mudah dimainkan.

6.2. Saran

Berikut saran-saran untuk pengembangan dan perbaikan sistem di masa yang akan datang. Diantaranya adalah sebagai berikut:

1. Memberikan *reward* kepada pemain ketika pemain selesai bermain, *reward* ini dapat berbentuk seperti suara tepuk tangan ketika pemain sudah selesai bermain satu permainan atau suara yang berisi kata-kata penyemangat ketika pemain gagal memainkan permainan dengan baik.
2. Pada saat menampilkan panduan bermain, lebih baik ditampilkan video permainan yang berisi panduan serta

contoh bermain sehingga dapat lebih memudahkan anak-anak dalam memahami permainan.

3. Bangun datar pada permainan “Match The Shape” dapat dikurangi jumlahnya sedikit dan letaknya dipadatkan ke tengah layar permainan agar lebih memudahkan anak dalam mengambil. Pada permainan “Find The Matching Animals” binatangnya dapat diperbesar untuk lebih mudah digenggam oleh pemain.
4. Pada saat uji coba lebih baik jika dilakukan beberapa kali. Dua skenario yang dilakukan lebih baik diterapkan 3-4 kali pada anak yang sama pada waktu yang berbeda, hal ini dilakukan untuk melihat perkembangan lebih lanjut pada pemahaman pemain serta kemampuan motorik halus dari pemain.

DAFTAR PUSTAKA

- [1] G. Zhu, S. Cai, Y. Ma and E. Liu, "A series of Leap Motion-based Matching Games for Enhancing the Fine Motor Skills of Children with Autism," *IEEE 15th International Conference on Advanced Learning*, 2015.
- [2] E. Hasnita and T. R. Hidayanti, "Terapi Okupasi Perkembangan Motorik Halus Anak Autisme," 2014.
- [3] Y. Chang, S. Chen and J. Huang, "A Kinect-based system for physical rehabilitation: A pilot Study for young adults with motor disabilities," *Research in developmental disabilities*, vol. 32, no. 6, pp. 2566-2570, 2011.
- [4] Wikipedia, "Interaksi Manusia-Komputer," Wikipedia, 14 April 2016. [Online]. Available: https://id.wikipedia.org/wiki/Interaksi_manusia-komputer. [Accessed 25 May 2016].
- [5] A. Nalwan, *Pemrograman Animasi dan Game Profesional*, Jakarta: Elex Media Komputindo, 1998.
- [6] Leap Motion Inc., "Leap Motion: Introducing the Skeletal Tracking Model," Leap Motion Inc., 2013. [Online]. Available: https://developer.leapmotion.com/documentation/objc/devguide/Intro_Skeleton_API.html?proglang=objc. [Accessed 16 November 2015].
- [7] Microsoft, "Microsoft studio- development tools," Microsoft, [Online]. Available: <http://www.visualstudio.com/>. [Accessed 9 april 2014].
- [8] Unity, "Create The Games You Love With Unity," Unity Technologies, 2014. [Online]. Available: <http://unity3d.com/unity>. [Accessed 17 September 2015].
- [9] Blender, "about-blender," Blender, [Online]. Available: <http://www.blender.org/about/>. [Accessed 10 april 2014].

- [10] Majalah Anaku, "Serba Serbi Terapi Autisme," Majalah Anaku, 10 01 2014. [Online]. Available: <http://www.anaku.net/serba-serbi-terapi-autisme.html>. [Accessed 08 06 2016].

BIODATA PENULIS



Penulis, Ahmad Fathoriq Fauzi, lahir di kota Jakarta pada tanggal 8 Maret 1994. Penulis dibesarkan di kota Jakarta, DKI Jakarta. Penulis menempuh pendidikan formal di TK Al-Azhar Rawamangun (1999-2000), SDI Al-Azhar 13 Rawamangun (2000-2006), SMPI Al-Azhar 12 Rawamangun (2006-2009), SMA Labschool Jakarta (2009-2012). Pada tahun 2012, penulis melanjutkan pendidikan S1 jurusan Teknik Informatika Fakultas Teknologi Informasi di Institut Teknologi Sepuluh Nopember Surabaya, Jawa Timur. Di jurusan Teknik Informatika, penulis mengambil bidang minat Interaksi Grafika dan Seni atau biasa disingkat menjadi IGS dan memiliki ketertarikan di bidang *Front End Web*, Sistem Enterprise, dan pembuatan *Game*. Penulis beberapa kali ikut dalam Lomba-lomba yang ada di luar kampus, seperti mengikuti Intel RealSense App Challenge yang diadakan oleh Fakultas Ilmu Komputer Universitas Indonesia, penulis juga mengikuti kompetisi yang diadakan oleh Bank Mandiri yaitu Mandiri Young Leaders, dan juga lomba-lomba lainnya. Penulis dapat dihubungi melalui alamat email ahmad.fathoriq@gmail.com